(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 September 2014 (12.09.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/136594 A1

(51) International Patent Classification:
G09C 1/00 (2006.01)

(21) International Application Number:
PCT/JP2014/054246

(22) International Filing Date:
17 February 2014 (17.02.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2013-045574 7 March 2013 (07.03.2013) JP
2014-017413 31 January 2014 (31.01.2014) JP

(71) Applicant: **CANON KABUSHIKI KAISHA** [JP/JP]; 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP).

(72) Inventors: **KARAKI, Seiga**; C/O CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP). **YAMADA, Shinya**; C/O CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP).

(74) Agents: **ABE, Takuma** et al.; C/O CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP).

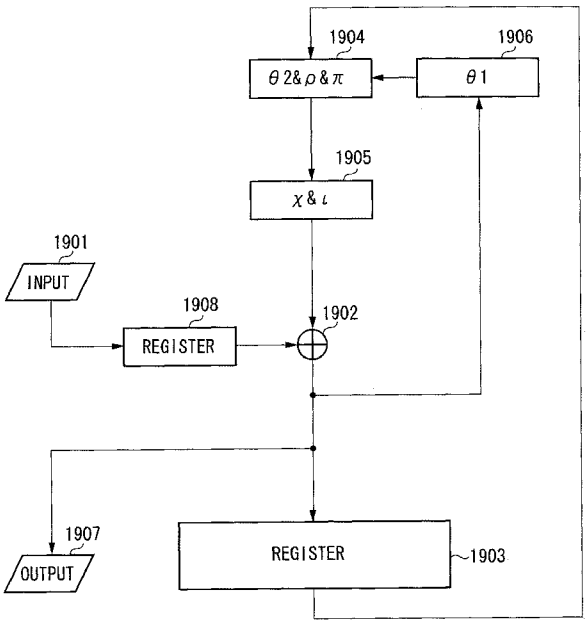
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

[Continued on next page]

(54) Title: HASH VALUE GENERATING DEVICE

[Fig. 16]



(57) Abstract: A hash value generating device for generating a hash value based on the KECCAK algorithm includes a θ processing unit, a ρ processing unit, a π processing unit, a χ processing unit, and an ι processing unit for performing processing of five steps θ, ρ, π, χ, and ι, included in round processing of the KECCAK algorithm. The π processing unit receives input of data in units of planes and outputs data in units of sheets.

WO 2014/136594 A1

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

DESCRIPTION

Title of Invention

HASH VALUE GENERATING DEVICE

Technical Field

[0001] The present invention relates to technique for generating a hash value.

Background Art

[0002] A hash value, which is calculated by using a cryptographic hash algorithm, is utilized for checking data alteration. It has been already verified that Secure Hash Algorithm 1 (SHA-1), which is a cryptographic hash algorithm (cryptographic hash), is not capable of securing safety. It has been pointed out that SHA-2 family (SHA-224, SHA-256, SHA-384, and SHA-512) may lack security. Therefore, National Institute of Standards and Technology (NIST) asked the public to come up with a new algorithm to establish a next-generation cryptographic hash algorithm (SHA-3). Then, the KECCAK algorithm ("The KECCAK reference", Version 3.0, January 14, 2011, (http://keccak.noekeon.org/Keccak-reference-3.0.pdf)) was assigned as the SHA-3 in December 2012.

[0003] The SHA-3 outputs a cryptographic hash value of a fixed length from an input message (data) of any length. In the KECCAK algorithm, a permutation function is used, and in the permutation function, round processing of five sequential

steps (θ, ρ, π, χ, and ι) is repeated twenty-four times. The round processing is performed on data called a "state" data piece having a length of 1600 bits.

[0004] A lot of results of preceding processing has to be once stored in a memory for the θ processing and the π processing out of the five steps of the round processing. Therefore, a lot of results of preceding processing has to be once stored in a memory twice within one-time round processing, and thus speedup has been difficult.

Summary of Invention

[0005] The present invention is directed to a technique to improve throughput for generating hash values.

[0006] According to an aspect of the present invention, a hash value generating device for generating a hash value based on KECCAK algorithm includes a θ processing means, a ρ processing means, a π processing means, a χ processing means, and an ι processing means for performing processing of five steps θ, ρ, π, χ, and ι included in round processing of the KECCAK algorithm, wherein the π processing unit receives input of data in units of planes and outputs data in units of sheets.

[0007] Further features of the present invention will become apparent from the following description of exemplary embodiments with reference to the attached drawings.

Brief Description of Drawings

[0008]

[Figs. 1A, 1B, and 1C] Figs. 1A, 1B, and 1C are diagrams illustrating the KECCAK algorithm.

[Figs. 2A, 2B, 2C, 2D, 2E, and 2F] Figs. 2A, 2B, 2C, 2D, 2E, and 2F are diagrams illustrating data structures.

[Figs. 3A and 3B] Figs. 3A and 3B are diagrams illustrating processing in step θ.

[Figs. 4A, 4B, and 4C] Figs. 4A, 4B, and 4C are diagrams illustrating processing in step ρ.

[Figs. 5A and 5B] Figs. 5A and 5B are diagrams illustrating processing in step π.

[Fig. 6] Fig. 6 is a diagram illustrating processing in step χ.

[Fig. 7] Fig. 7 is a diagram illustrating processing in step ι.

[Fig. 8] Fig. 8 is a diagram illustrating round constants in step ι.

[Figs. 9A, 9B, and 9C] Figs. 9A, 9B, and 9C are diagrams illustrating an overview of round processing R'.

[Figs. 10A, 10B, and 10C] Figs. 10A, 10B, and 10C are diagrams illustrating processing in step ρ'.

[Fig. 11] Fig. 11 is a diagram illustrating processing in step θ1.

[Fig. 12] Fig. 12 is a diagram illustrating processing in step θ2.

[Figs. 13A, 13B, and 13C] Figs. 13A, 13B, and 13C are diagrams illustrating processing in step θ2'.

[Figs. 14A, 14B, and 14C] Figs. 14A, 14B, and 14C are

operation timing charts.

[Fig. 15] Fig. 15 is a diagram illustrating a schematic configuration of an implementation example when processing is performed on a lane as a unit by the KECCAK algorithm.

[Fig. 16] Fig. 16 is a diagram illustrating a schematic configuration of an implementation example of the KECCAK algorithm according to a first exemplary embodiment.

[Fig. 17] Fig. 17 is a diagram illustrating more detail of the configuration illustrated in Fig. 16.

[Fig. 18] Fig. 18 is a diagram illustrating an implementation example of a register.

[Fig. 19] Fig. 19 is a diagram illustrating an implementation example of a θ2 & ρ & π circuit.

[Fig. 20] Fig. 20 is a diagram illustrating an implementation example of a χ & ι circuit.

[Fig. 21] Fig. 21 is a diagram illustrating a schematic configuration of an implementation example of the KECCAK algorithm according to a second exemplary embodiment.

[Fig. 22] Fig. 22 is a diagram illustrating an implementation example of a θ2& ρ′ & χ & ι circuit.

[Figs. 23A and 23B] Figs. 23A and 23B are diagrams illustrating characteristics of processing in step π.

[Figs. 24A, 24B, 24C, 24D, and 24E] Figs. 24A, 24B, 24C, 24D, and 24E are diagrams illustrating sheet-plane unit conversion performed by the register.

[Fig. 25] Fig. 25 is a diagram illustrating an implementation example of a π circuit.

Description of Embodiments

[0009] Various exemplary embodiments, features, and aspects of the invention will be described in detail below with reference to the drawings.

[0010] As a hash value generating device according to a first exemplary embodiment of the present invention, a device configured to generate a hash value of SHA-3 (KECCAK algorithm) will be hereinafter described as an example. In the description below, a specific data length or a specific bit value may be provided, but the present invention is not limited to the specific length or value.

[0011] First, the KECCAK algorithm will be described. Note that, specifications in more detail can be found in "The KECCAK reference", Version 3.0, January 14, 2011, (http://keccak.noekeon.org/Keccak-reference-3.0.pdf).

[0012] Fig. 1A is a diagram illustrating a whole of the KECCAK algorithm. In Fig. 1A, message blocks 101 (m_1 to m_t) are illustrated. The message blocks 101 (m_1 to m_t) are generated by dividing an input message, for which a hash value is generated, into units of 1024-bit blocks.

[0013] As illustrated in Fig. 1A, all bits of initial values 102 and 103 are zeros in the present embodiment. Here, an example where all bits of initial values are zeros is described as an example, but the present invention is not limited to this example. The length of the initial value 102 is 1024 bits, which is the same as that of the message blocks described above, and

the total length of the initial values 102 and 103 is 1600 bits. A bitwise exclusive OR (XOR) operator 104 is also illustrated. That is, the XOR operator 104 calculates exclusive OR for each bit of the two 1024-bit input data pieces and outputs the results as a 1024-bit data piece.

[0014] A KECCAK-f 105, which is a permutation function, receives two input data pieces and outputs two data pieces. The detail of the KECCAK-f 105 will be described below with reference to Fig. 1B. A cut-out section 106 cuts out a necessary size from the 1024-bit input data pieces, and outputs the cut out data. A cryptographic hash value (i.e., hash value) 107 is calculation results of this algorithm.

[0015] Fig. 1B is a diagram illustrating an overview of the KECCAK-f 105, which is a permutation function. Round processing R 201 is performed twenty-four times. The detail of the round processing R will be described below. Input data pieces 202 and 203 are illustrated. The length of the input data piece 202 is 1024 bits. The total length of the input data pieces 202 and 203 is 1600 bits. The two input data pieces 202 and 203 are coupled and then input to the round processing R 201. Output data pieces 204 and 205 are illustrated. The length of the output data piece 204 is 1024 bits. The total length of the output data pieces 204 and 205 is 1600 bits.

[0016] Fig. 1C is a diagram illustrating an overview of the round processing R 201. As described above, for the round processing R 201, the lengths of the input data piece and the output data piece are both 1600 bits. In the round processing

R 201, processing of five steps (θ processing unit 301, ρ processing unit 302, π processing unit 303, χ processing unit 304, and ι processing unit 305) to be described below is sequentially performed on the input data piece to generate the output data piece.

[0017] Data structures used in the round processing of the KECCAK algorithm and the above five steps will be described in detail below.

[0018] Fig. 2A is a diagram illustrating a "state", which is a data structure upon input/output of the round processing R 201. As described above, both of the input data piece and the output data piece have 1600-bit length. Each of these 1600-bit data pieces is expressed as a rectangular parallelepiped having a width (x axis direction) of five bits, a height (y axis direction) of five bits, and a depth (z axis direction) of sixty-four bits in three-dimensional arrangement. The rectangular parallelepiped data structure is called a "state". A 1600-bit data piece is allocated to the state structure expressed as a rectangular parallelepiped in the order of the z axis direction, the x axis direction, and the y axis direction. The detail will be described below with reference to Fig. 2F.

[0019] Fig. 2B is a diagram illustrating a data structure "plane". The plane structure is expressed as a planar structure that is parallel to the x-z plane and that has a width of five bits, a height of one bit, and a depth of sixty-four bits. That is, the above "state" structure can be considered as five plane

structures that are stacked in the y axis direction.

[0020] Fig. 2C is a diagram illustrating a data structure "sheet". The sheet structure is expressed as a planar structure that is parallel to the y-z plane and that has a width of one bit, a height of five bits, and a depth of sixty-four bits. That is, the above "state" structure can be considered as five sheet structures arranged horizontally in line in the x axis direction.

[0021] Fig. 2D is a diagram illustrating a data structure "lane". The lane structure is expressed as a linear structure that is parallel to the z axis and that has a width of one bit, a height of one bit, and a depth of sixty-four bits. That is, the above "state" structure can be considered as twenty-five lane structures gathered along the x-y plane. Fig. 2F is a diagram illustrating the order of twenty-five lanes included in one state structure.

[0022] Fig. 2E is a diagram illustrating a data structure "column". The column structure is expressed as a linear structure that is parallel to the y axis and that has a width of one bit, a height of five bits, and a depth of one bit. That is, the above "sheet" structure can be considered as sixty-four column structures arranged in line in the z axis direction.

[0023] In the first exemplary embodiment, a case where the input data piece is 1600 bits are described, but the present invention is not limited to the case. In addition, an example where data of the state structure is handled as a rectangular parallelepiped data structure having a width (x axis direction)

of five bits, a height (y axis direction) of five bits, and a depth (z axis direction) of sixty-four bits will be described, but the present invention is not limited thereto. For example, an input data piece may have 800 bits, and the state structure data may be handled as a rectangular parallelepiped data structure having a width of five bits, a height of five bits, and a depth of thirty-two bits.

[0024] Further, the plane structure, the sheet structure, the lane structure, and the column structure can be modified according to the respective numbers of bits in the width (x axis direction), in the height (y axis direction), and in the depth (z axis direction) of the state structure. More specifically, when the state structure data has m bits in the x axis direction, n bits in the y axis direction, and s bits in the z axis direction, the plane structure is a planar structure having m bits in the x axis direction, one bit in the y axis direction, and s bits in the z axis direction. The sheet structure is a planar structure having one bit in the x axis direction, n bits in the y axis direction, and s bits in the z axis direction. The lane structure is a linear structure having one bit in the x axis direction, one bit in the y axis direction, and s bits in the z axis direction. The column structure is a linear structure having one bit in the x axis direction, n bits in the y axis direction, and one bit in the z axis direction.

[0025] Next, a method of forming an input data piece for the first round processing R 201 from the input data pieces 202 and 203 that have been input to the KECCAK-f 105 will be described.

First, the input data pieces 202 and 203 are coupled in this order to form a 1600-bit data block. Next, the 1600-bit data block is divided into units of sixty-four bits to form twenty-five lanes. Last, the twenty-five lanes are arranged in the order illustrated in Fig. 2F along the x-y plane to build one state. The thus generated state structure is input to the round processing R 201. A method of forming the output data pieces 204 and 205 from an output data piece of the twenty-fourth round processing R 201 is similar, and thus the description thereof is not provided.

[0026] Next, five steps (steps θ, ρ, π, χ, and ι) included in the round processing R 201 will be described. In each of the steps, the data structure of an input data piece and an output data piece is the state structure.

[0027] Fig. 3A is a diagram illustrating processing in the step θ (θ processing unit 301). The step θ is processing of adding the sum of two columns to each bit, the two columns being adjacent to the bit. The θ processing unit calculates each bit of the output state as the sum of three values obtained from the input state: "a value of a bit at the same position"; "the sum of bits of a column at a position of -1 in the x axis direction"; and "the sum of bits of a column at a position of +1 in the x axis direction and -1 in the z axis direction". Here, the sum means the sum on GF(2), and the result will be the same as that of the exclusive OR operation. The processing can be expressed by the following expression.

$$a'[x][y][z] \leftarrow a[x][y][z] + \sum_{y'=0}^{4} a[x-1][y'][z] + \sum_{y'=0}^{4} a[x+1][y'][z-1]$$

In the expression, x is 0 to 4, y is 0 to 4, z is 0 to 63.

[0028] Fig. 3B is a diagram illustrating processing in the step θ upon calculation of a bit in an end part (x = 0, for example). In order to calculate a bit at x = 0, "a column at a position of -1 in the x axis direction" corresponds to a column opposite in the state, that is, "the column at a position of x = 4". As described above, a coordinate beyond the state is regarded as a position that is opposite in the state. That is, coordinate values are cyclically shifted in the same state. This rule is similarly applied to x coordinate, y coordinate, and z coordinate and to four other steps.

[0029] Figs. 4A, 4B, and 4C are diagrams illustrating processing in the step ρ (ρ processing unit 302). The step ρ is processing of shifting values of respective bits in the z axis direction. More specifically, the ρ processing unit 302 cyclically shifts values in each lane of the state in the z direction by the specified number of bits as illustrated in Fig. 4A and outputs the shifted values. The number of bits by which the values are shifted in each lane is previously determined as the number illustrated in Fig. 4B. Note that, in order to perform the ρ processing, a holding section previously holds a table listing shifting amounts as illustrated in Fig. 4C and the ρ processing unit 302 performs the ρ processing using the table being held.

[0030] Figs. 5A and 5B are diagrams illustrating processing in the step π (π processing unit 303). The step π is processing of rearranging each of the respective bits in the x-y plane, that is, processing of rearranging twenty-five lanes in a single state. Note that the x-y plane may be also referred to as a "slice". More specifically, when respective lanes in the input state are numbered as illustrated in the upper part of Fig. 5A, the output state is illustrated in the lower part thereof. Note that, in order to perform the π processing, the holding section previously holds a table listing rearrangement destinations as illustrated in Fig. 5B and the π processing unit 303 performs the π processing using the table being held.

[0031] Fig. 6 is a diagram illustrating processing in the step χ (χ processing unit 304). The step χ is processing of converting a bit using bits in a line in the x axis direction (also referred to as a "row"), and each bit in the output row is derived based on three bits in the same input row. More specifically, setting is made such that when a bit at a position of +1 in the x axis direction from each bit of the input row is 0 and a bit at a position of +2 in the x axis direction from the bit is 1, the χ processing unit 304 inverts the value of the each corresponding bit of the output row.

[0032] Fig. 7 is a diagram illustrating processing in the step ι (ι processing unit 305). The step ι is processing of adding a round constant to each bit. Fig. 8 is a diagram illustrating round constants used in the step ι. In the step ι, the ι processing unit 305 performs exclusive OR (XOR) on a

bit line of a lane at x = y = 0 with a round constant (64-bit value) predetermined for each round. More specifically, the ι processing unit 305 calculates bitwise exclusive OR of a 64-bit value of a lane at x = y = 0 (when a bit at z = 63 is MSB and a bit at z = 0 is LSB) and a round constant illustrated in Fig. 8. Then, the ι processing unit 305 sets the result as a bit line of a lane at x = y = 0 in the output state.

[0033] From the processing contents of the above respective steps (steps θ, ρ, π, χ, and ι), it can be understood that there are following limitations regarding start of the processing of the respective steps.

· In the step θ, the θ processing unit 301 uses a sheet data piece at -1 and a sheet data piece at +1 in the x axis direction to calculate each lane in the state. Therefore, when the first three sheets are completed, that is, when the θ processing unit 301 receives twenty-three lanes out of the twenty-five lanes from a preceding stage, the θ processing unit 301 can start the processing in the step θ.

· The step ρ is calculation for each of lanes independent of each other. Therefore, when one lane of calculation results of the preceding stage (step θ) is output, the ρ processing unit 302 can start the processing in the step ρ.

· In the step π, respective lanes in a state are rearranged. Therefore, when one whole state of calculation results of the preceding stage (step ρ) is output, that is, when twenty-five lanes are output, the π processing unit 303 can start the processing in the step π.

· In the step χ, in calculation of each lane in a state, the χ processing unit 304 uses a lane at +1 in the x axis direction and a lane at +2 in the x axis direction. Therefore, upon receiving three lane data pieces, the χ processing unit 304 can start the processing in the step χ.

· The step ι is calculation for each of lanes independent of each other. Therefore, when one lane of calculation results of the preceding stage (step χ) is output, the ι processing unit 305 can start the processing in the step ι.

[0034] In other words, in the steps θ, π, and χ, start of processing has to wait until the steps at the respective preceding stages output calculation results of twenty-three lanes, twenty-five lanes, and three lanes respectively. As described above, particularly the processing of the two steps θ and π can be started when a long time has passed after the start of processing of their preceding stages.

[0035] This means that throughput can be improved when the starting time of the step θ or the step π can be hastened. In addition, throughput can be improved by performing the processing in units of planes or units of sheets instead of the unit of lanes in each step.

[0036] Next, round processing R′ 801 will be described. The round processing R′ 801 is processing used in the present embodiment and designed such that the result is the same as that of the round processing R 201.

[0037] Fig. 9A is a diagram illustrating an overview of the round processing R′ 801. The round processing R′ 801 is

designed such that the processing result is the same as that of the round processing R 201. In the round processing R' 801, processing of six steps (performed by a θ1 processing unit 802, a θ2 processing unit 803, a ρ processing unit 804, a π processing unit 805, a χ processing unit 806, and an ι processing unit 807) is performed on the input data piece to generate an output data piece.

[0038] Here, the ρ processing unit 804, the π processing unit 805, the χ processing unit 806, and the ι processing unit 807 are respectively similar to the ρ processing unit 302, the π processing unit 303, the χ processing unit 304, and the ι processing unit 305 in the round processing R 201. The θ1 processing unit 802 and the θ2 processing unit 803 are obtained by dividing the θ processing unit 301 in the round processing R 201.

[0039] Because the ρ processing, the χ processing, and the ι processing in the round processing R' 801 are similar to those in the round processing R 201, the description thereof is not provided.

[0040] The π processing unit 805 is similar to the π processing unit 303 in the round processing R 201. However, the π processing unit 805 does not perform the processing after holding a state data piece, but outputs a sheet data piece from a plane data piece having been input. The detail will be described below.

[0041] The θ1 processing, and the θ2 processing will be described below.

[0042] Fig. 11 is a diagram illustrating processing in the step θ1. The step θ1 corresponds to the first half operation of the step θ. In the step, column sum calculation processing is performed. More specifically, the processing is for calculating, for each column, the sum (to be referred to as a θ intermediate value) of two values: "the sum of bits in a column at a position of -1 in the x axis direction" and "the sum of bits in a column at a position of +1 in the x axis direction and -1 in the z axis direction". After receiving five sheet data pieces, the θ1 processing unit 802 outputs a θ intermediate value of one bit for each column that totals up to θ intermediate values of five times sixty-four bits. A structure for all of the θ intermediate values will be expressed as a planar structure that is parallel to the x-z plane and that has a width of five bits, a height of one bit, and a depth of sixty-four bits.

[0043] Fig. 12 is a diagram illustrating the processing in the step θ2. The step θ2 corresponds to the second half operation of the step θ, and is a step of performing column sum addition processing. That is, in the step θ2, the θ2 processing unit 803 adds the θ intermediate values calculated in the step θ1 to respective bits.

[0044] From the processing contents of the above respective steps (step θ1 and step θ2), it can be understood that there are following limitations regarding start of the processing of the respective steps.

· The step θ1 is calculation of the sum, and thus, the θ1

processing unit 802 updates a θ intermediate value in the process of calculation every time when each sheet in the state is input. Therefore, when one sheet data piece of calculation results of the preceding stage is output, the θ1 processing unit 802 can start the processing in the step θ1.

· In the step θ2, the θ2 processing unit 803 adds a θ intermediate value calculated in the step θ1 in calculation of each plane in the state. Since the step θ1 has been completed at the time of starting the step θ2, the θ2 processing unit 803 can start to output the processing result of the step θ2 when one plane data piece is output from the preceding stage (register).

· The step ρ is calculation for each of lanes independent of each other. Therefore, when the θ2 processing unit 803 outputs one plane data piece of calculation results of the preceding stage (step θ2), the ρ processing unit 804 can start the processing in the step ρ.

· In the step π, the π processing unit 805 rearranges respective lanes in a state. However, from input of one plane, output of one sheet can be obtained. Therefore, when the ρ processing unit 804 outputs one plane data piece of the calculation results of the preceding stage (step ρ), the π processing unit 805 can start the processing in the step π.

· In the step χ, in calculation of each lane in a state, the χ processing unit 806 uses a lane at +1 in the x axis direction and a lane at +2 in the x axis direction. Therefore, the χ processing unit 806 can start the processing in the step χ after receiving three sheet data pieces.

· The step ι is calculation for each of lanes independent of each other. Therefore, when the χ processing unit 806 outputs one sheet data piece of calculation results of the preceding stage (step χ), the ι processing unit 807 can start the processing in the step ι.

[0045] Since the π processing unit 805 receives input of a plane data piece and outputs a sheet data piece in the step π, the need of holding a state data piece can be eliminated and throughput can be improved.

[0046] In addition, since the processing is performed in units of planes in the steps θ2, ρ and π, and in units of sheets in the steps χ and ι, throughput can be improved.

[0047] There will be described below a configuration in which the π processing unit 805 receives input of plane data pieces and outputs sheet data pieces, and the round processing is performed in units of planes in the steps θ2, ρ, and π, and performed in units of sheets in the steps χ and ι.

[0048] Fig. 16 is a diagram illustrating a schematic configuration of an implementation example of the KECCAK algorithm according to the first exemplary embodiment. In this example, a lane structure out of an input data piece 1901 is input as a unit. However, the order of input data pieces is previously determined to be in sequential in the y direction: (x, y) = (0, 0), (0, 1), (0, 2), ... A register 1908 holds four lane structure data pieces that are the input data pieces 1901 and outputs a sheet structure data piece as a unit when the register 1908 receives the fifth lane structure data piece.

Note that, the register 1908 may be configured to output the first sheet structure data piece when the register 1908 holds data enabling generation of at least the first sheet structure data piece. An output data piece 1907 is output in units of sheet structure data pieces when calculation is completed.

[0049] An exclusive OR (XOR) operator 1902 calculates exclusive OR of a message block and internal data each time of performing the round processing twenty-four times. A register 1903 holds the whole of the internal data. The register 1903 holds input data pieces of the sheet structure that are input and outputs output data pieces of the plane structure. A circuit 1904 is for the processing in the steps θ2, ρ, and π. Data input to the circuit 1904 is data of the plane structure, and data output from the circuit 1904 is data of the sheet structure. The detail will be described below. A circuit 1905 is provided for the processing in the steps χ and ι. A circuit 1906 is provided for processing in the step θ1.

[0050] Fig. 17 is a diagram illustrating the configuration of Fig. 16 in more detail. An input data piece 2001 is the same as the input data piece 1901 in Fig. 16. A register 2009 holds at least the first sheet structure data piece out of the input data piece 2001 and outputs a sheet structure data piece as a unit. The register 2009 is the same as the register 1908 in Fig. 16. The register 2009 outputs one sheet structure data piece as a unit at every clock, and outputs sheet structure data pieces in ascending order of the x coordinate. A multiplexer 2002 outputs an input data piece as is when the multiplexer 2002

calculates exclusive OR of the input data piece and the internal state and outputs zero in other cases.

[0051] A register 2004 holds the whole of the internal data similarly to the register 1903 illustrated in Fig. 16. A circuit 2005 calculates in the steps θ2, ρ, and π (hereinafter, referred to as a θ2 & ρ & π circuit 2005). As described above, to the θ2 & ρ & π circuit 2005, plane structure data pieces are input in ascending order of the y coordinate (in the order of y = 0, 1, 2, 3, and 4), and the θ2 & ρ & π circuit 2005 outputs sheet structure data pieces in ascending order of the x coordinate (in the order of x = 0, 1, 2, 3, and 4).

[0052] A circuit 2006 is for the processing in the steps χ and ι (hereinafter, referred to as a χ & ι circuit 2006) and outputs one sheet structure data piece of the calculation results as a unit. A multiplexer 2007 outputs zero for initialization upon start of hash value calculation and outputs data in the process of calculation in other cases.

[0053] A circuit 2008 is for the processing in the step θ1 (hereinafter, referred to as a θ1 circuit 2008) and outputs an intermediate value (θ intermediate value) of five times sixty-four bits when the circuit 2008 receives input of five sheet data pieces.

[0054] Fig. 18 is a diagram illustrating an implementation example of the register 2004. A combination circuit 2102 divides one sheet data piece that is an input data piece 2101 into five lane data pieces and outputs the five lane data pieces. A multiplexer 2103 alternately switches the shift direction to

the vertical direction and the horizontal direction for even times of the round and odd times of the round (in Fig. 18, all symbols of the same shape as the multiplexer 2103 perform similarly thereto). A register 2104 is of a mesh configuration of five times five stages for storing information of one lane in one stage. The register 2104 receives input of five lane data pieces and outputs five lane data pieces (data of one sheet or one plane) at every clock. A combination circuit 2105 outputs data of five lanes (five lanes including R0, R1, R2, R3, and R4 or five lanes including R0, R5, R10, R15, and R20) having been output from the last stage of the register 2104 is output as one plane data piece. An output data piece 2106 is illustrated.

[0055] Figs. 23A and 23B are diagrams illustrating characteristics of the processing in the step π.

[0056] Fig. 23A illustrates an example of data before the processing in the step π is performed. For ease of understanding of positions to which the respective lanes are rearranged by the π processing, sequential numbers are allocated to the respective lanes. Fig. 23B illustrates an example of data after the processing in the step π is performed. In this diagram, it can be seen that the bits of one plane data piece at y = 0 (a data piece 231 in Fig. 23A) are all rearranged to positions in one sheet data piece at x = 0 (a data piece 232 in Fig. 23B). The data pieces at y = 1, 2, 3, and 4 are rearranged similarly. That is, when the processing in the step π is performed, each plane data piece at y = i (i = 0, 1, 2, 3, and

4) is output as one sheet data piece at x = i.

[0057] By utilizing the characteristics of the processing in the step π as described above, the need of holding all data pieces (i.e., five plane structure data pieces) in a register for the processing in the step π is eliminated, and thus from input of one plane, one sheet output can be obtained.

[0058] Figs. 24A, 24B, 24C, 24D, and 24E are diagrams illustrating sheet-plane unit conversion performed by the register 2004. An input data piece for the register 2004 is one sheet data piece for one clock as described above. Figs. 24A, 24B, and 24C are explanation diagrams of the register 2004 when a sheet data piece is input at the first clock, when the sheet data piece is input at the second clock, and when the sheet data piece is input at the fifth clock respectively. As illustrated in Figs. 24A, 24B, and 24C, the register 2004 receives input of five sheet data pieces (one state data piece) over five clocks in the y direction (x direction). Fig. 24D is an explanation diagram of the register 2004 when the register 2004 outputs the first plane data piece after receiving input of five sheet data pieces. Fig. 24E is an explanation diagram of the register 2004 when the register 2004 outputs the second plane data piece. The register 2004 outputs each of five lane data pieces at every clock in the x direction (y direction) after receiving input of five sheet data pieces to provide one plane data piece. The register 2004 performs sheet-plane conversion in the above flow.

[0059] Note that, data is input to the register 2004 while

switching the input direction between the x axis direction and the y axis direction alternately. For example, five plane data pieces are input in the y axis direction, and then five sheet data pieces are input in the x axis direction.

[0060] Fig. 19 is a diagram illustrating an implementation example of the θ2 & ρ & π circuit 2005. The θ2 & ρ & π circuit 2005 calculates output data pieces using θ intermediate values that the circuit 2008 has previously calculated for a target plane. Since the processing includes the processing in the step π, output data pieces has the sheet structure.

[0061] As described above, the θ2 & ρ & π circuit 2005 receives input of one plane data piece at every clock as an input data piece 2201. An input data piece 2202 is input from the θ1 circuit 2008 and corresponds to a θ intermediate value.

[0062] The processing of an exclusive OR (XOR) operator 2203 corresponds to the operation of the step θ2 as described above. A logic circuit (ρ circuit) 2204 performs the operation of the step ρ. A logic circuit (π circuit) 2205 performs the operation of the step π and can output one sheet from input of one plane. An output data piece 2206 is one sheet data piece that is output at every clock.

[0063] Fig. 25 is a diagram illustrating an implementation example of the π circuit 2205. As described above, the π circuit 2205 receives input of one plane data piece at every clock as an input data piece 2601. A combination circuit 2602 divides one plane data piece that is the input data piece 2601 into five lane data pieces and outputs the five lane data pieces.

[0064] A combination circuit 2603 rearranges in the step π. The combination circuit 2603 rearranges five lane data pieces aligned in the x axis direction to five lane data pieces aligned in the y axis direction according to the table of Fig. 5B and outputs the five lane data pieces.

[0065] A combination circuit 2604 outputs five lane data pieces as one sheet data piece. An output data piece 2605 is one sheet data piece at every clock.

[0066] Fig. 20 is a diagram illustrating an implementation example of the χ & ι circuit 2006. The χ & ι circuit 2006 calculates an output data piece by using a target sheet data piece and two sheet data pieces at positions of +1 and +2 in the x axis direction from the target sheet data piece.

[0067] As described above, the χ & ι circuit 2006 receives input of one sheet data piece at every clock as an input data piece 2301. A multiplexer 2302 outputs the input data piece 2301 as is during the first five clocks from the start of the processing and outputs data from a register 2304 during the following two clocks.

[0068] A register 2303 having a two-stage configuration stores information of one sheet in one stage. The register 2304 having a two-stage configuration stores information of one sheet in one stage. The register 2304 stores information of sheets at x = 0 and x = 1.

[0069] A combination circuit 2305 is a logic circuit configured to perform the operations of the above steps χ and ι. An output data piece 2306 of one sheet is output as a unit.

[0070] As described above, the θ2 & ρ & π circuit 2005 is for the processing in units of planes, and the χ & ι circuit 2006 is for the processing in units of sheets. Thus, on the path from the output of the θ2 & ρ & π circuit 2005 to the input of the χ & ι circuit 2006, only a combination circuit is connected. More specifically, a latch circuit is not included on the path. Therefore, a data piece can pass through the path within one clock.

[0071] As described above, according to the first exemplary embodiment, the π processing unit 805 receives input of a plane and outputs a sheet, and processing in a round is performed using a plane and a sheet as units. Thus, the processing in the round can be pipelined.

[0072] In addition, as described above, according to the first exemplary embodiment, a unit of input processing at the start of a round is the plane, and a unit of output processing at the end of a round is the sheet. However, since plane-sheet unit conversion is performed from the input to the output of the register 2004, the unit of input processing can be units of planes at the start of a next round.

[0073] Note that, an input unit to the θ1 circuit 2008 is the sheet structure in the first exemplary embodiment, but the input unit may be the plane structure. In that case, after receiving five plane data pieces, the θ1 circuit 2008 outputs θ intermediate values of five times sixty-four bits. However, when the input unit to the θ1 circuit 2008 is the sheet structure, the θ1 circuit 2008 can sequentially calculate a θ intermediate

value at every input, which is an advantageous effect.

[0074] Fig. 14A is an operation timing chart of the respective modules of the implementation example according to the first exemplary embodiment. The θ2 & ρ & π circuit 2005 and the χ & ι circuit 2006 are configured for pipeline processing. Since the θ1 circuit 2008 can calculate a θ intermediate value from input of five sheet data pieces, it takes five clocks. Two clocks after the θ2 & ρ & π circuit 2005 outputs one sheet data piece, the χ & ι circuit 2006 can output one sheet data piece. That is, the χ & ι circuit 2006 can start the processing upon receiving three sheet data pieces. Thus, the θ2 & ρ & π circuit 2005 and the χ & ι circuit 2006 can operate in parallel. In addition, it takes seven clocks for one-time round processing.

[0075] Hereinafter, an implementation example in which processing is performed on a lane data piece as a unit by the algorithm according to specifications will be described for comparison with the implementation example of the above first exemplary embodiment.

[0076] Fig. 15 is a diagram illustrating a schematic configuration of the implementation example when processing is performed on a lane as a unit by the KECCAK algorithm according to specifications. The processing of the five steps (θ, ρ, π, χ, and ι) is similar to that described above, and thus the description thereof is not provided.

[0077] The KECCAK-f receives one lane data piece (data having a length of sixty-four bits) from an input data piece 1801 at every clock. The KECCAK-f receives lane data pieces in one

state data piece in the order illustrated in Fig. 2F.

[0078] An Exclusive OR processing unit 1802 is an operator that calculates exclusive OR of a message block and an internal data piece every time when the round processing is performed twenty-four times.

[0079] A register 1803 holds the whole of the internal data expressed as a state data piece. A processing block (π circuit) 1804 performs the step π. However, as described above, the processing in the step π can be performed only after the processing in the step ρ is completed. A processing block (θ circuit) 1805 performs the step θ, and a processing block (ρ circuit) 1806 performs the step ρ.

[0080] A processing block (χ circuit) 1807 performs the step χ, and a processing block (ι circuit) 1808 performs the step ι. A multiplexer 1809 outputs data having been input from the processing block 1806 in the first half of round processing and outputs data from the processing block 1808 in the second half thereof. The KECCAK-f outputs an output data piece 1810 of one lane when calculation is completed.

[0081] Fig. 14B is an operation timing chart of the respective modules when processing is performed on a lane as a unit by the algorithm according to specifications. A pair of the θ circuit 1805 and the ρ circuit 1806 and a pair of the χ circuit 1807 and the ι circuit 1808 operate in different time-periods and do not operate at the same time. It takes fifty-one clocks for one-time round processing.

[0082] As can be seen from comparison of Fig. 14A and Fig.

14C, throughput of the processing can be improved by using the configuration of the implementation example according to the first exemplary embodiment.

More specifically, the followings can be said.

· The two processing circuits: the θ2 & ρ & π circuit 2005 and the χ & ι circuit 2006 operate in parallel, and thus utilization efficiency of the circuits can be improved.

· One-time round processing can be performed within a smaller number of clocks (less time).

[0083] As a hash value generating device according to a second exemplary embodiment of the present invention, a device configured to generate a hash value of SHA-3 (KECCAK algorithm) will be hereinafter described as an example. In the description below, a specific data length or a specific bit value may be provided, but the present invention is not limited to the specific values. The KECCAK algorithm and the data structures are similar to those of the first exemplary embodiment, and thus, points different from the first exemplary embodiment will be described.

[0084] Round processing R' 901 will be described. The round processing R' 901 is processing used in the present embodiment and designed such that the result is the same as that of the round processing R 201. However, processing contents of the round processing R' 901 are different from the specifications of the KECCAK algorithm.

[0085] Fig. 9B is a diagram illustrating an overview of the

round processing R' 901. The round processing R' 901 is designed such that the processing result is the same as that of the round processing R 201. In the round processing R' 901, processing of six steps is performed (by a θ1 processing unit 902, a π processing unit 903, a θ2' processing unit 904, a ρ' processing unit 905, a χ processing unit 906, and an ι processing unit 907) on an input data piece to generate an output data piece.

[0086] Note that, the π processing unit 903, the χ processing unit 906, and the ι processing unit 907 performs processing similar to those performed by the π processing unit 303, the χ processing unit 304, and the ι processing unit 305 of the round processing R 201. The ρ' processing unit 905 performs processing of shifting values of respective bits in the z axis direction similarly to the ρ processing unit 302 of the round processing R 201, but the number of bits by which the values are shifted is different. The θ1 processing unit 902 and the θ2' processing unit 904 are obtained by dividing the θ processing unit 301 in the round processing R 201.

[0087] Since the χ processing and the ι processing in the round processing R' 901 are similar to those in the round processing R 201, the description thereof is not provided.

[0088] The π processing unit 903 is similar to the π processing unit 303 in the round processing R 201. However, the π processing unit 903 does not perform the processing after a state data piece is held, but receives a plane data piece and then outputs a sheet data piece. The detail is similar to the first exemplary embodiment.

[0089] The θ1 processing of the round processing R' 901 is similar to the θ1 processing of the round processing R' 801, and thus the description thereof is not provided. The ρ' processing and the θ2' processing will be described below.

[0090] Fig. 10A is a diagram illustrating processing in the step ρ' (ρ' processing unit 905). In the step ρ', the ρ' processing unit 905 performs processing of cyclically shifting a value of each bit in the z axis direction similarly to the step ρ. However, the number of bits by which the values are cyclically shifted in each lane is different from that of the step ρ, and is illustrated in Fig. 10B. Note that, in order to perform the ρ' processing, a holding section previously holds a table listing shifting amounts as illustrated in Fig. 10C and the ρ' processing unit 905 performs the ρ' processing using the table being held. This table is determined in consideration of the π processing. The detail will be described below.

[0091] In order to describe that the processing result of the round processing R' 901 and that of the round processing R 201 are the same, first, there will be described that the processing result of the round processing R 201 and the processing result of round processing R'' 911 are the same.

[0092] Fig. 9C is a diagram of the round processing R'' 911. In the round processing R'' 911, processing of five steps is performed (by a θ processing unit 912, a π processing unit 913, a ρ' processing unit 915, a χ processing unit 916, and an ι processing unit 917) on the input data piece to generate an output data piece. Here, the θ processing unit 912, the π

processing unit 913, the χ processing unit 916, and the ι processing unit 917 are respectively similar to the θ processing unit 301, the π processing unit 303, the χ processing unit 304, and the ι processing unit 305 of the round processing R 201. The ρ′ processing unit 915 is similar to the ρ′ processing unit 905 of the round processing R′ 901.

[0093] When the round processing R 201 is compared with the round processing R′′ 911, they are different in a point that the π processing unit 913 and the ρ′ processing unit 915 perform the processing in this order in the round processing R′′ 911 while the ρ processing unit 302 and the π processing unit 303 perform the processing in this order in the round processing R 201.

[0094] Here, in the step ρ of the round processing R 201, the ρ processing unit 302 shifts values in the z axis direction according to rules determined for respective lanes, and the π processing unit 303 rearranges the respective lanes. On the other hand, in the round processing R′′ 911, the π processing unit 913 rearranges the respective lanes (processing in the step π), and thereafter the ρ′ processing unit 915 shifts values in the z axis direction according to rules determined for the respective lanes in consideration of the rearrangement processing (processing in the step ρ′). More specifically, in the round processing R′′ 911, the step π is performed before the step ρ′, but the shifting amount by which values are shifted in the z axis direction by the ρ′ processing unit 915 is changed in consideration of the processing in the step π, whereby the

processing result of the round processing R'' 911 becomes the same as that of the round processing R 201.

[0095] Fig. 10C is a table listing shifting amounts for respective lanes used in the step ρ'.

[0096] A method of generating the table illustrated in Fig. 10C will be specifically described. First, the round processing R 201 will be considered. In the round processing R 201, the ρ processing unit 302 and the π processing unit 303 perform the processing in this order. The numbers in Fig. 4B are shifting amounts in the step ρ. For example, the shifting amount for a lane at the position of x = 0 and y = 4 is eighteen bits. Next, the lane rearrangement by the π processing is confirmed using Figs. 5A and 5B. It can be seen that the π processing unit 303 moves the lane at the position of x = 0 and y = 4 to the position of x = 4, y = 2.

[0097] Next, the round processing R'' 911 will be considered. In the round processing R'' 911, the π processing unit 913 and the ρ' processing unit 915 perform the processing in this order. Since the π processing is performed before the ρ' processing, a lane for which the ρ' processing unit 915 should shift values by eighteen bits is a lane at the position of x = 4, y = 2. Therefore, the number at the position of x = 4, y = 2 in Fig. 10B is eighteen. Shifting amounts of the other lanes can be similarly obtained to be the other numbers in Fig. 10B.

[0098] That is, the table listing the shifting amounts for the respective lanes used in the step ρ' illustrated in Fig. 10C is a table determined in consideration of the rearrangement

processing of the π processing.

[0099] Next, there will be described that the processing result of the round processing R'' 911 is the same as that of the round processing R' 901.

[0100] Note that, the π processing unit 903, the ρ' processing unit 905, the χ processing unit 906, and the ι processing unit 907 respectively perform the processing similarly to the π processing unit 913, the ρ' processing unit 915, the χ processing unit 916, and the ι processing unit 917 of the round processing R'' 911. The θ1 processing unit 902 and the θ2' processing unit 904 are obtained by dividing the θ processing unit 912.

[0101] When the round processing R'' 911 is compared with the round processing R' 901, they are different in a point that the θ processing unit 912 and the π processing unit 913 perform the processing in this order in the round processing R'' 911 while the θ1 processing unit 902, the π processing unit 903, and the θ2' processing unit 904 perform the processing in this order in the round processing R' 901.

[0102] Here, in the round processing R'' 911, the step θ is a step of adding the sum of two columns to each bit, the two columns being adjacent to the bit, and the step π is a step of rearranging the respective lanes. On the other hand, in the round processing R' 901, the θ1 processing unit 902 calculates the sum of two columns that are adjacent to each bit (processing in the step θ1). Then, the π processing unit 913 rearranges the respective lanes (processing in the step π), and the θ2'

processing unit 904 adds the sum of the columns to a bit in consideration of the rearrangement of the respective lanes (processing in the step θ2').

[0103] Fig. 13A is a diagram illustrating processing in the step θ2'. The step θ2' corresponds to the operation of the second half of the step θ and is a step of performing column sum addition processing. That is, the step θ2' is a step of adding θ intermediate values calculated in the step θ1 to the respective bits.

[0104] However, it should be noted that the step π has been already performed before the step θ2'. More specifically, in the step θ of the round processing R'' 911 (i.e., the step θ of the round processing R 201), an x coordinate of each bit and an x coordinate of a θ intermediate value used for calculation of the bit is the same. However, in the step θ2' of the round processing R' 901, an x coordinate of each bit and an x coordinate of a θ intermediate value used for calculation of the bit is different and the x coordinate is determined in consideration of the rearrangement of the respective lanes in the step π. The x coordinates of θ intermediate values used for calculation of respective bits are illustrated in Fig. 13B. Note that, a holding section previously holds a table in Fig. 13C providing x coordinates of θ intermediate values used for calculation of respective bits in the θ2' processing, and the θ2' processing unit 904 performs the θ2' processing using the table being held.

[0105] A method of generating the table illustrated in Fig. 13C will be specifically described. First, the round

processing R'' 911 will be considered. The x coordinates of θ intermediate values needed to calculate respective bits in the step θ are the same as the x coordinates of the respective bits. For example, a bit at the position of x = 0, y = 4 is calculated using a θ intermediate value at the position of x = 0 in the step θ. Next, the lane rearrangement in the step π is confirmed using Figs. 5A and 5B. It can be seen that the π processing unit 913 moves the bit at the position of x = 0, y = 4 to the position of x = 4, y = 2.

[0106] Next, the round processing R' 901 will be considered. Since the π processing unit 903 has already performed the step π when the θ2' processing unit 904 performs the step θ2', it can be seen that an x coordinate of a θ intermediate value needed for calculation of a bit at the position of x = 4, y = 2 in the step θ2' is x = 0. Therefore, a number at the position of x = 4, y = 2 out of the numbers provided in Fig. 13B becomes zero. The x coordinates of θ intermediate values for other bits can be similarly obtained to be the other numbers in Fig. 13B.

[0107] That is, the table in Fig. 13C providing the x coordinates of θ intermediate values when the θ2' processing unit 904 performs the step θ2' is a table determined in consideration of the rearrangement processing of the π processing.

[0108] As described above, the processing result of the round processing R 201 and that of the round processing R'' 911 are the same. In addition, the processing result of the round processing R'' 911 and the processing result of the round

processing R' 901 are the same. Therefore, the processing result of the round processing R' 901 and the processing result of the round processing R 201 are the same.

[0109] From the processing contents of the above respective steps (steps θ1, θ2', and ρ'), it can be understood that there are following limitations regarding start of the processing of the respective steps.

· In the step θ1, the θ1 processing unit 902 calculates the sum, and thus, the θ1 processing unit 902 updates a θ intermediate value in the process of calculation every time when each plane in the state is input. Therefore, when one plane data piece of calculation results of the preceding stage is output, the θ1 processing unit 902 can start the processing in the step θ1.

· In the step θ2', the θ2' processing unit 904 adds a θ intermediate value calculated in the step θ1 in calculation of each plane in the state. Since the step θ1 has been completed at the time of starting the step θ2', the θ2' processing unit 904 can start to output the processing result of the step θ2' when one plane data piece of calculation results of the preceding stage (step π) is output.

· The step ρ' is calculation for each of lanes independent of each other. Therefore, when the θ2' processing unit 904 outputs one plane data piece of calculation results of the preceding stage (step θ2'), the ρ' processing unit 905 can start the processing in the step ρ'.

[0110] That is, in the steps θ1, θ2', and ρ', processing can be started when one plane data piece out of calculation results

of a step of the preceding stage is output.

[0111] In addition, the processing contents of the steps π, χ, and ι of the second exemplary embodiment are the same as those described in the first exemplary embodiment. Thus, it can be understood that there are following limitations regarding start of the processing of the respective steps.

· In the step π, respective lanes in a state are rearranged. However, the π processing unit 903 can output one sheet data piece from input of one plane data piece. Therefore, when the preceding stage (step θ1) outputs one plane of calculation results, the π processing unit 903 can start the processing in the step π.

· In the step χ, in calculation of each lane in a state, the χ processing unit 906 uses a lane at +1 and a lane at +2 in the x axis direction. Therefore, when the ρ' processing unit 905 outputs one plane data piece of calculation results of the preceding stage (step ρ'), the χ processing unit 906 can start the processing in the step χ.

· The step ι is calculation for each of lanes independent of each other. Therefore, when one plane data piece of calculation results of the preceding stage (step χ) is output, the ι processing unit 907 can start the processing in the step ι.

[0112] Since a plane data piece is input and a sheet data piece is output in the step π, the need of holding a state data piece can be eliminated and throughput can be improved.

[0113] In addition, the π processing unit 903 performs processing in units of planes, and the round processing R' 901

is used instead of the round processing R 201. Therefore, throughput can be improved. Hereinafter, a configuration in which plane data pieces are input to the step π and sheet data pieces are output therefrom, and the round processing is performed in units of planes will be described.

[0114] Fig. 21 is a diagram illustrating a configuration of an implementation example of the KECCAK algorithm according to the second exemplary embodiment. Here, the order of input data pieces 2401 is previously determined to be sequential in the x direction: (x, y) = (0, 0), (1, 0), (2, 0), ... A register 2409 holds four lane structure data pieces that are the input data pieces 2401 and outputs a plane structure data piece as a unit when the register 2409 receives the fifth lane structure data piece. Note that, the register 2409 may be configured to output the first plane structure data piece when the register 2409 holds data enabling generation of at least the first plane structure data piece. When calculation is completed, an output data piece 2410 is output in units of the sheet structures.

[0115] A multiplexer 2402 outputs an input data piece as is when the multiplexer 2402 calculates exclusive OR of the input data piece and the internal state and outputs zero in other cases.

[0116] A circuit 2404 calculates in the step π (hereinafter, referred to as a π circuit 2404). The π circuit 2404 receives input of a plane structure data piece and outputs a sheet structure data piece similarly to the π circuit 2205 illustrated in Fig. 19. More specifically, plane structure data pieces are

input in ascending order of the y coordinate (in the order of y = 0, 1, 2, 3, and 4), and outputs sheet structure data pieces in ascending order of the x coordinate (in the order of x = 0, 1, 2, 3, and 4).

[0117] A register 2405 holds the whole of the internal data. Since the register 2405 is similar to the register 2004 of the first exemplary embodiment, the description thereof is not provided.

[0118] A circuit 2406 performs calculation in the steps θ2', ρ', χ, and ι (hereinafter, referred to as a θ2' & ρ' & χ & ι circuit 2406) and outputs one plane structure data piece of the calculation results as a unit. A multiplexer 2407 outputs zero upon start of calculation of a hash value for initialization and outputs data in the process of calculation in other cases.

[0119] A circuit 2408 is for the processing in the step θ1 (hereinafter, referred to as a θ1 circuit 2408) and outputs an intermediate value (θ intermediate value) of five times sixty-four bits when the circuit 2408 receives five plane data pieces.

[0120] Fig. 22 is a diagram illustrating an implementation example of the θ2' & ρ' & χ & ι circuit 2406. The θ2' & ρ' & χ & ι circuit 2406 calculates output data pieces using θ intermediate values that the circuit 2408 has previously calculated for a target plane.

[0121] The θ2' & ρ' & χ & ι circuit 2406 receives input of one plane data piece at every clock as an input data piece 2501. An input data piece 2502 is input from the θ1 circuit 2408, and

corresponds to θ intermediate values.

[0122] The processing by an exclusive OR (XOR) operator 2503 corresponds to the operation of the step θ2′ described above. A combination circuit 2504 is a logic circuit configured to actually perform the operations of the above steps ρ′, χ, and ι and to outputs one plane data piece at every clock. An output data piece 2505 is one plane data piece that is output at every clock.

[0123] As described above, the π circuit 2404 performs processing in units of planes and outputs a sheet structure data piece, the register 2405 converts the sheet structure data piece to a plane structure, and the θ2′ & ρ′ & χ & ι circuit 2406 performs processing in units of planes. On the path from the output of the θ2′ & ρ′ & χ & ι circuit 2406 and to the input of the π circuit 2404, only a combination circuit is connected. More specifically, a latch circuit is not included on the path. Therefore, a data piece can pass through the path within one clock.

[0124] Note that, an input unit to the θ1 circuit 2408 is the plane structure in the second exemplary embodiment, but the input unit may be the sheet structure. In that case, after receiving input of five sheet data pieces, the θ1 circuit 2408 outputs θ intermediate values of five times sixty-four bits.

[0125] Fig. 14B is an operation timing chart of the respective modules of the implementation example according to the second exemplary embodiment. The output from the θ2′ & ρ′ & χ & ι circuit 2406 to the π circuit 2404 is configured for

pipeline processing. The processing of each of the all steps in the round processing can be started when one plane data piece is input. That is, all steps in the round processing can be performed in parallel. In addition, it takes five clocks for one-time round processing.

[0126] Hereinafter, an implementation example in which processing is performed on a lane data piece as a unit by the algorithm according to specifications will be described for comparison with the implementation example of the above second exemplary embodiment. The implementation example in which processing is performed on a lane data piece as a unit by the algorithm according to the specifications is similar to the contents described in the first exemplary embodiment, and thus the description thereof is not provided.

[0127] Fig. 14C is an operation timing chart of the respective modules of the implementation example in which processing is performed on a lane data piece as a unit according to specifications.

[0128] As can be seen from comparison of Fig. 14B and Fig. 14C, throughput of the processing can be improved by using the configuration of the implementation example according to the second exemplary embodiment.

More specifically, the followings can be said.

· The two processing circuits: the π circuit 2404 and the θ2′ & ρ′ & χ & ι circuit 2406 operate in parallel, and thus utilization efficiency of the circuits can be improved.

· One-time round processing can be performed within a smaller

number of clocks (less time).

[0129] As described above in the present embodiment, the conversion of processing unit during the θ1 processing can reduce time for holding data.

[0130] According to the exemplary embodiments, a technique capable of improving throughput for generating hash values can be provided.

[0131] Embodiments of the present invention can also be realized by a computer of a system or apparatus that reads out and executes computer executable instructions recorded on a storage medium (e.g., non-transitory computer-readable storage medium) to perform the functions of one or more of the above-described embodiment(s) of the present invention, and by a method performed by the computer of the system or apparatus by, for example, reading out and executing the computer executable instructions from the storage medium to perform the functions of one or more of the above-described embodiment(s). The computer may comprise one or more of a central processing unit (CPU), micro processing unit (MPU), or other circuitry, and may include a network of separate computers or separate computer processors. The computer executable instructions may be provided to the computer, for example, from a network or the storage medium. The storage medium may include, for example, one or more of a hard disk, a random-access memory (RAM), a read only memory (ROM), a storage of distributed computing systems, an optical disk (such as a compact disc (CD), digital versatile disc (DVD), or Blu-ray Disc (BD)™), a flash memory device, a

memory card, and the like.

[0132] While the present invention has been described with reference to exemplary embodiments, it is to be understood that the invention is not limited to the disclosed exemplary embodiments. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures and functions.

[0133] This application claims the benefit of Japanese Patent Applications No. 2013-045574 filed March 7, 2013 and No. 2014-017413 filed January 31, 2014, which are hereby incorporated by reference herein in their entirety.

CLAIMS

[Claim 1]

A hash value generating device comprising:

a θ processing means for performing θ processing included in round processing of Secure Hash Algorithm 3 (SHA-3 algorithm);

a ρ processing means for performing ρ processing included in the round processing;

a π processing means for performing π processing included in the round processing;

a χ processing means for performing χ processing included in the round processing; and

an ι processing means for performing ι processing included in the round processing,

wherein the π processing means receives input of data in units of planes and outputs data in units of sheets.

[Claim 2]

The hash value generating device according to claim 1, further comprising a holding means for holding data on which processing has been performed,

wherein the holding means receives input of data in units of sheets and outputs data in units of planes.

[Claim 3]

The hash value generating device according to claim 2, wherein

the holding means outputs one plane after having held five sheets.

[Claim 4]

The hash value generating device according to claim 3, wherein the θ processing means includes a θ1 processing means for performing column sum calculation processing and a θ2 processing means for performing column sum addition processing.

[Claim 5]

The hash value generating device according to claim 4, wherein the θ1 processing means performs the θ1 processing during a period in which five sheets are input to the holding means.

[Claim 6]

The hash value generating device according to claim 2, wherein the holding means holds data processed by the π processing means.

[Claim 7]

The hash value generating device according to claim 4, wherein in the round processing, the π processing means performs processing before the θ2 processing means and the ρ processing means perform processing.

[Claim 8]

The hash value generating device according to claim 7, wherein

the θ2 processing means, the ρ processing means, the χ processing means, and the ι processing means perform processing in units of planes.

[Claim 9]

The hash value generating device according to claim 2, wherein the holding means holds data processed by the ι processing means.

[Claim 10]

The hash value generating device according to claim 4, wherein the θ2 processing means and the ρ processing means perform processing in units of planes, and

wherein the χ processing means and the ι processing means perform processing in units of sheets.

[Claim 11]

The hash value generating device according to claim 1, further comprising an output means for outputting a hash value obtained by performing the round processing using the θ means, the ρ means, the π means, the χ means, and the ι means.

[Claim 12]

The hash value generating device according to claim 1, wherein the plane is data of a structure having m bits in an x axis direction, one bit in a y axis direction, and s bits in a z axis direction, and the sheet is data of a structure having one bit

in the x axis direction, n bits in the y axis direction, and s bits in the z axis direction.

[Claim 13]

The hash value generating device according to claim 1, wherein the θ processing means calculates a sum of bits along an x axis direction and adds the calculated sum to a predetermined bit,

wherein the ρ processing means shifts values of respective bits in a z axis direction,

wherein the π processing means rearranges values of respective bits in an x-y plane,

wherein the χ processing means converts a bit using bits in a line in the x axis direction, and

wherein the ι processing means adds predetermined values to respective bits.

[Claim 14]

A hash value generating device that performs round processing of hash algorithm in which a data piece of a structure having m bits in an x axis direction, n bits in a y axis direction, and s bits in a z axis direction is processed, the hash value generating device comprising:

a first processing means for calculating a sum of bits in the x axis direction and adding the calculated sum to a predetermined bit;

a second processing means for shifting a bit in the z axis direction;

a third processing means for rearranging respective bits in an x-y plane;

a fourth processing means for converting a bit using bits in a line in the x axis direction; and

a fifth processing means for adding predetermined values to respective bits,

wherein the third processing means receives input of data in units of structures each having m bits in the x axis direction, one bit in the y axis direction, and s bits in the z axis direction and outputs data in units of structures each having one bit in the x axis direction, n bits in the y axis direction, and s bits in the z axis direction.

[Claim 15]

The hash value generating device according to claim 14, further comprising a holding means for holding data on which processing has been performed,

wherein the holding means receives input of data in units of structures each having one bit in the x axis direction, n bits in the y axis direction, and s bits in the z axis direction and outputs data in units of structures having m bits in the x axis direction, one bit in the y axis direction, and s bits in the z axis direction.

[Claim 16]

The hash value generating device according to claim 15, wherein the holding means outputs one unit of data of the

structure having m bits in the x axis direction, one bit in the y axis direction, and s bits in the z axis direction after holding five units of data of the structure having one bit in the x axis direction, n bits in the y axis direction, and s bits in the z axis direction.

[Claim 17]

The hash value generating device according to claim 16, wherein the first processing means includes a sixth processing means for calculating the sum of bits in the x axis direction and a seventh processing means for adding the calculated sum to the predetermined bit.

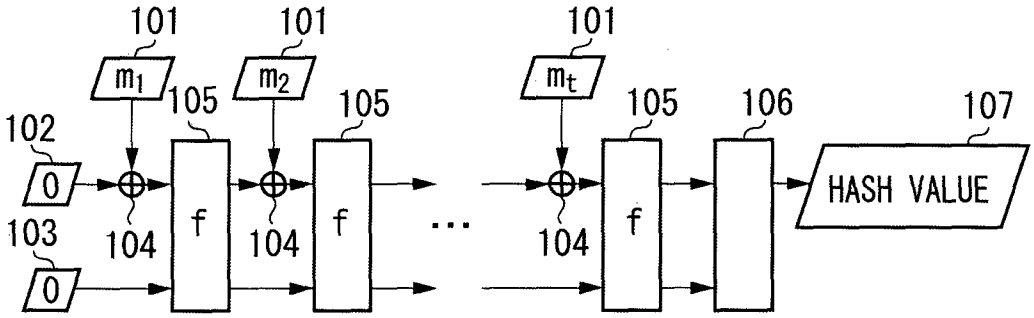
[Claim 18]

The hash value generating device according to claim 17, wherein the sixth processing means performs processing during a period in which the holding means receives five units of data of the structure having one bit in the x axis direction, n bits in the y axis direction, and s bits in the z axis direction.

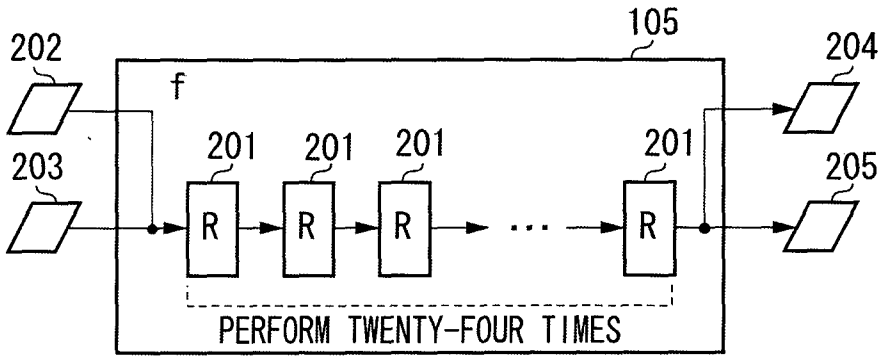
[Claim 19]

The hash value generating device according to claim 14, further comprising an output means for outputting a hash value obtained by performing the round processing using the first processing means, the second processing means, the third processing means, the fourth processing means, and the fifth processing means.

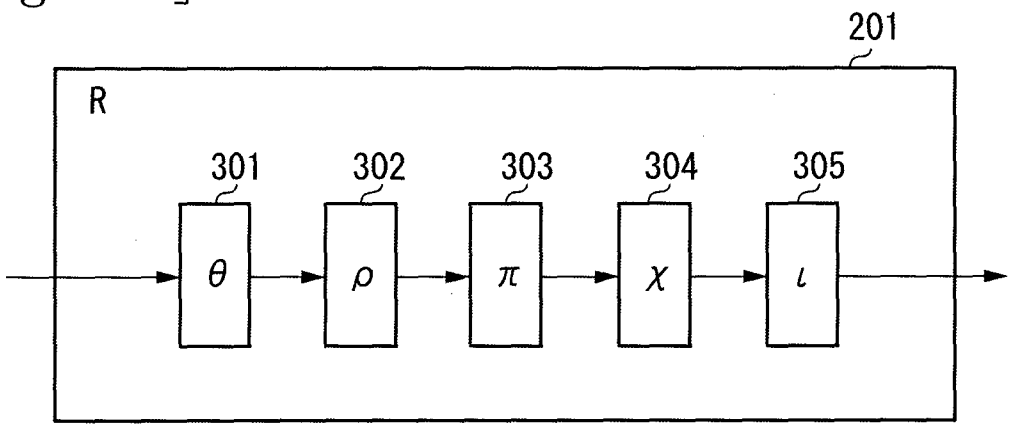
[Fig. 1A]



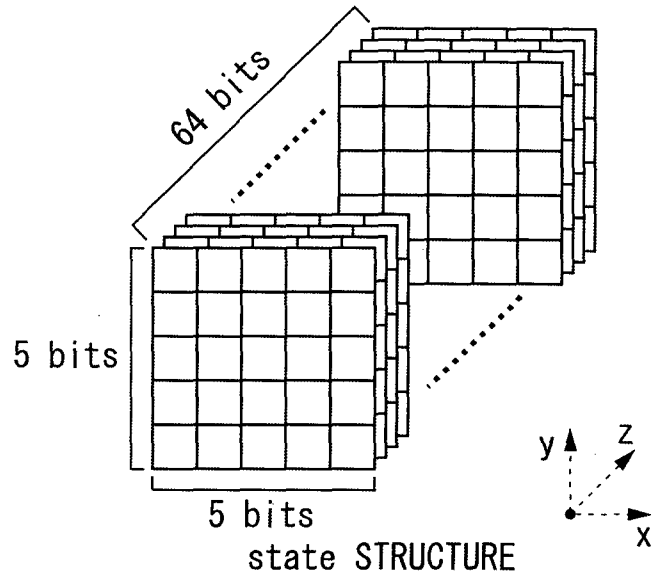
[Fig. 1B]



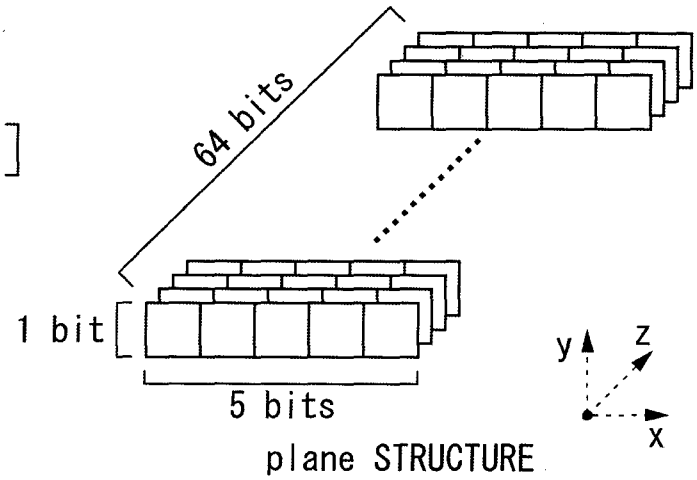
[Fig. 1C]



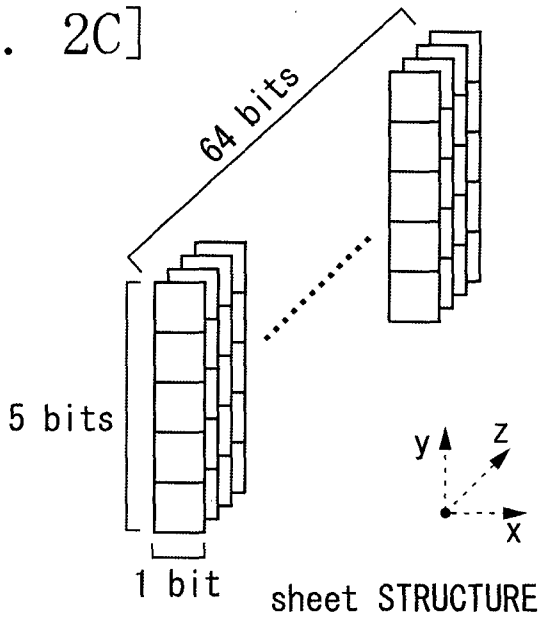
[Fig. 2A]



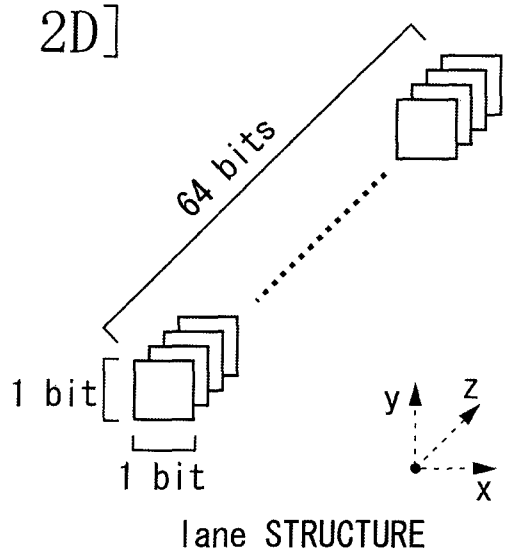
[Fig. 2B]



[Fig. 2C]

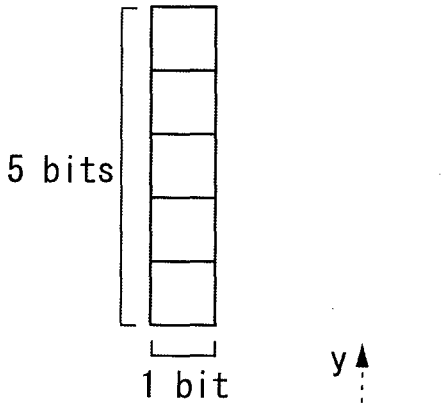


[Fig. 2D]



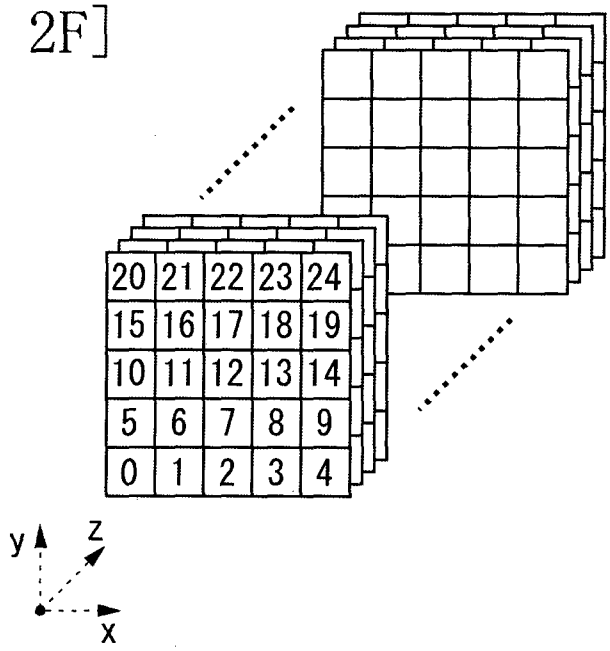
lane STRUCTURE

[Fig. 2E]

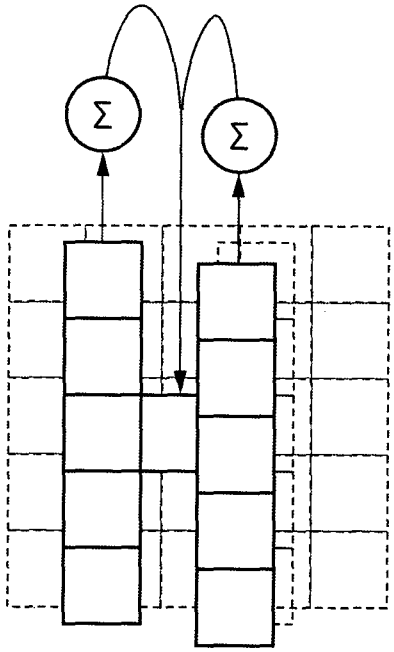


column STRUCTURE

[Fig. 2F]

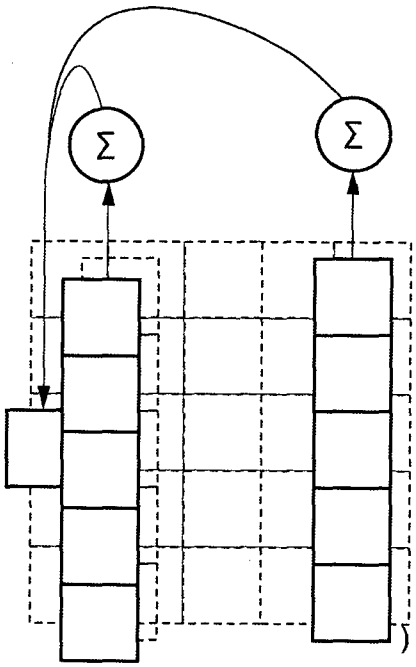


[Fig. 3A]



STEP θ

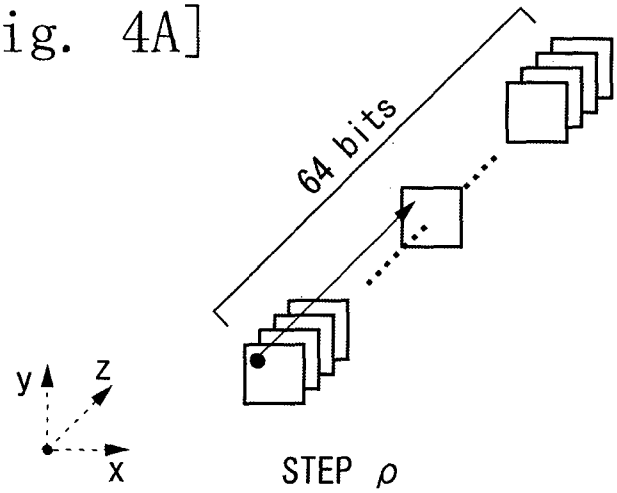
[Fig. 3B]



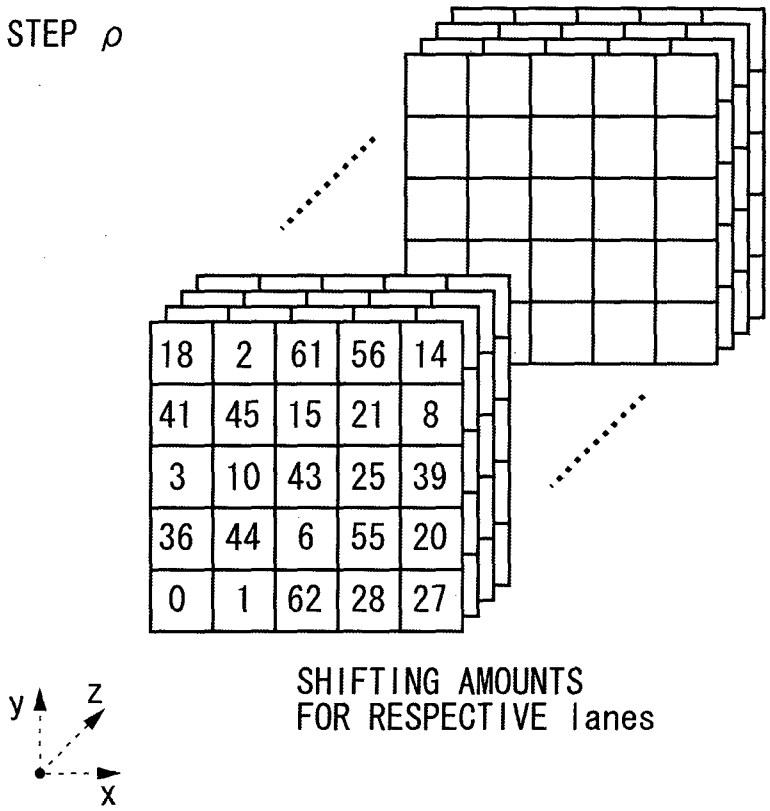
STEP θ

(UPON CALCULATION OF BIT AT X = 0)

[Fig. 4A]



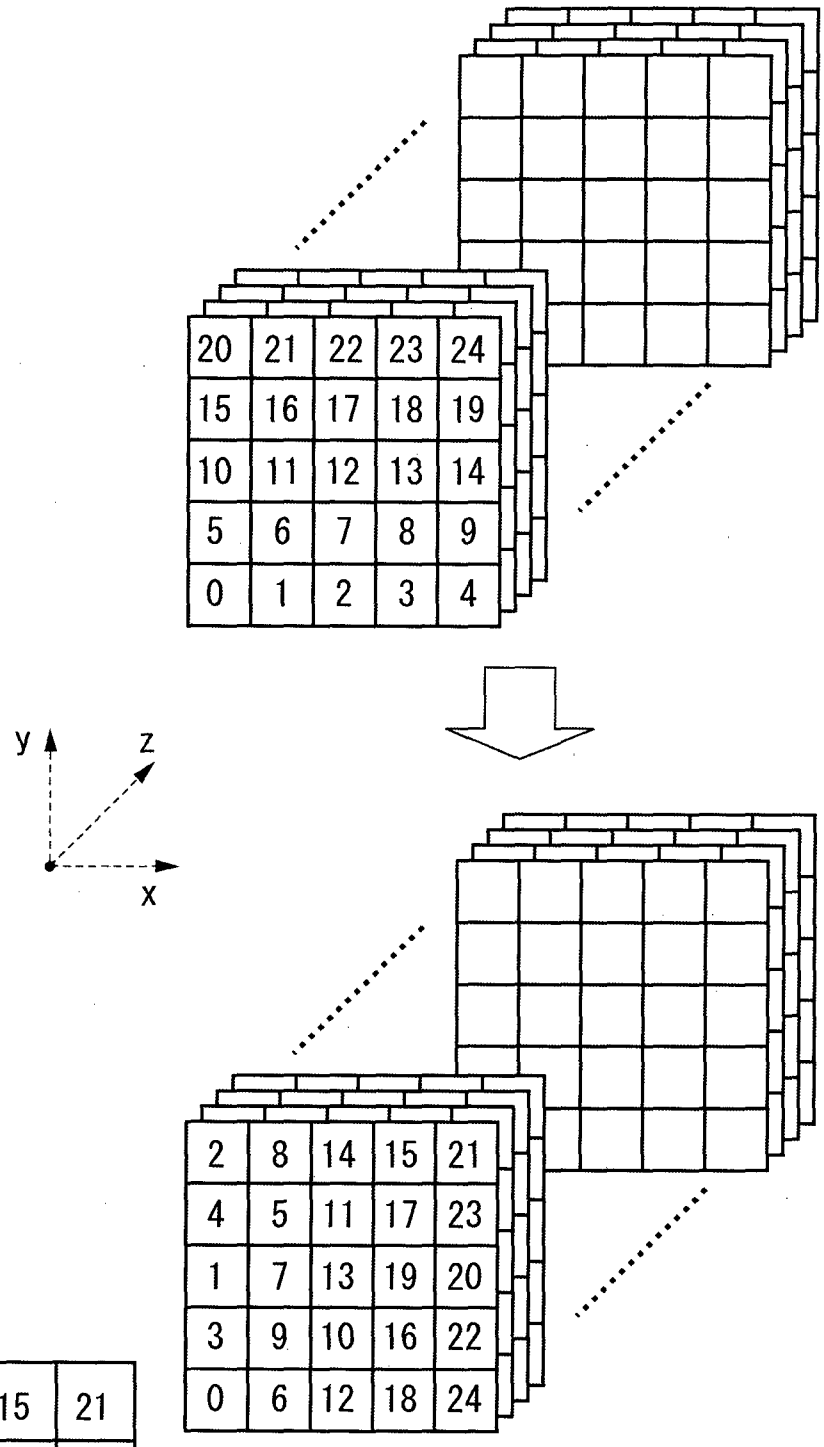
[Fig. 4B]



[Fig. 4C]

18	2	61	56	14
41	45	15	21	8
3	10	43	25	39
36	44	6	55	20
0	1	62	28	27

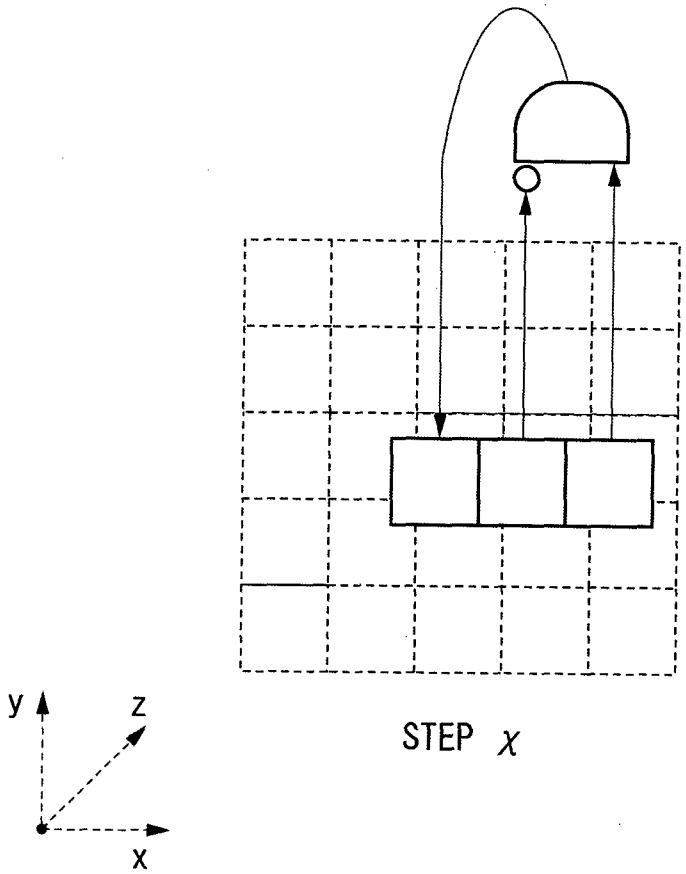
[Fig. 5A]



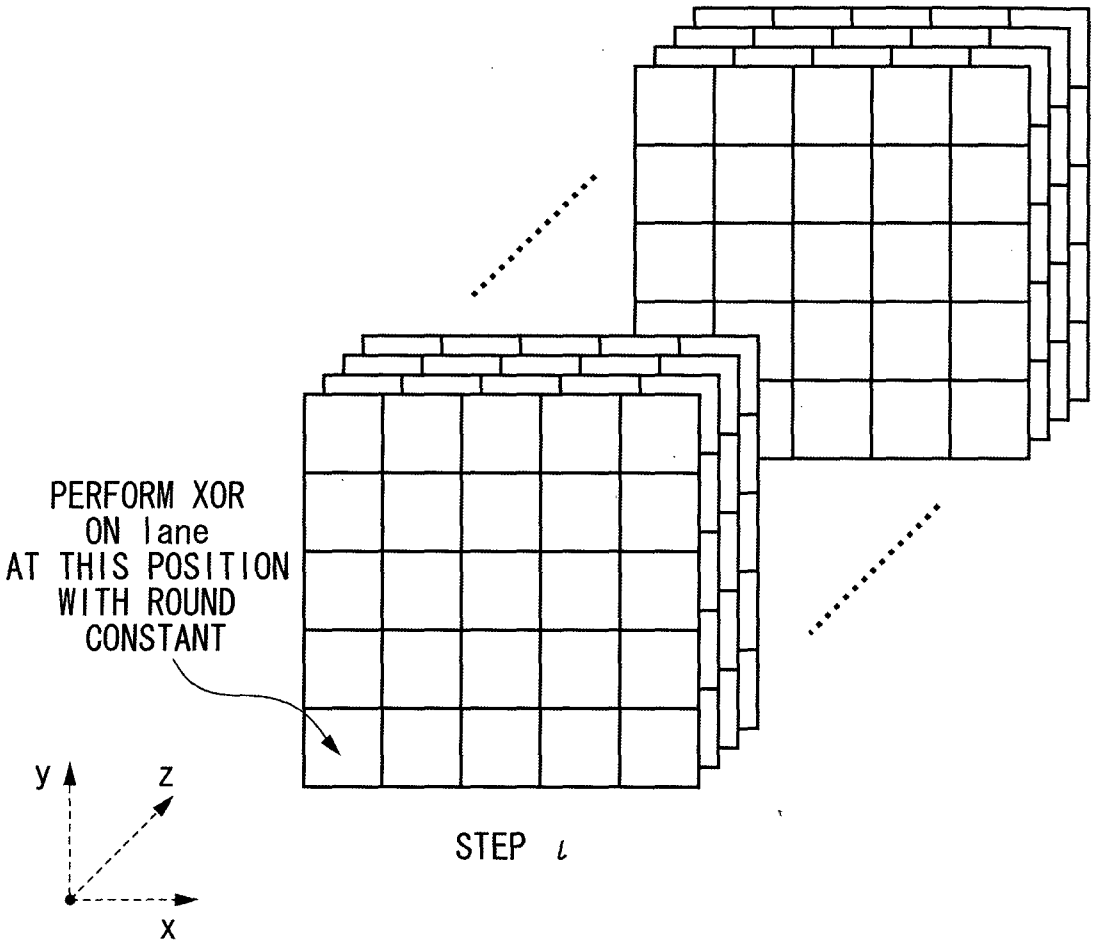
[Fig. 5B]

2	8	14	15	21
4	5	11	17	23
1	7	13	19	20
3	9	10	16	22
0	6	12	18	24

[Fig. 6]



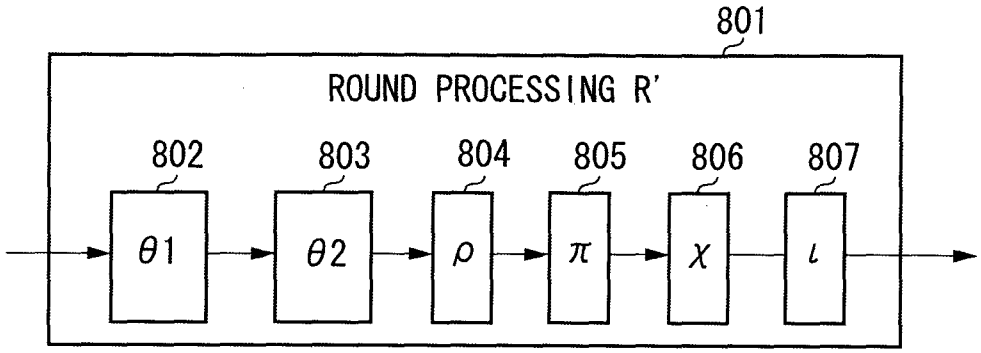
[Fig. 7]


PERFORM XOR
ON lane
AT THIS POSITION
WITH ROUND
CONSTANT
y
z
x
STEP ι

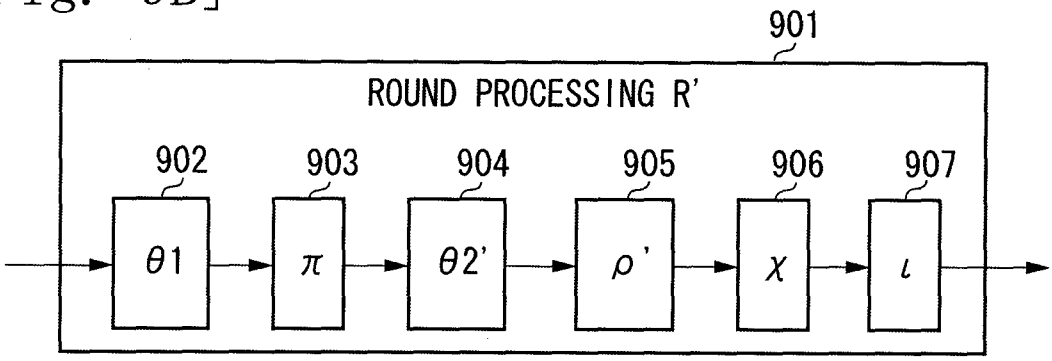
[Fig. 8]

NUMBER OF TIMES OF ROUND	ROUND CONSTANT
1	0x0000000000000001
2	0x0000000000008082
3	0x800000000000808A
4	0x8000000080008000
5	0x000000000000808B
6	0x0000000080000001
7	0x8000000080008081
8	0x8000000000008009
9	0x000000000000008A
10	0x0000000000000088
11	0x0000000080008009
12	0x000000008000000A
13	0x000000008000808B
14	0x800000000000008B
15	0x8000000000008089
16	0x8000000000008003
17	0x8000000000008002
18	0x8000000000000080
19	0x800000000000800A
20	0x800000008000000A
21	0x8000000080008081
22	0x8000000000008080
23	0x0000000080000001
24	0x8000000080008008

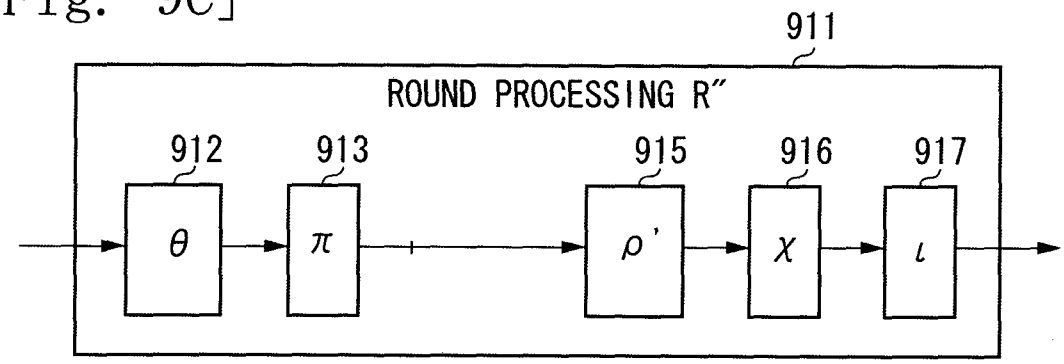
[Fig. 9A]



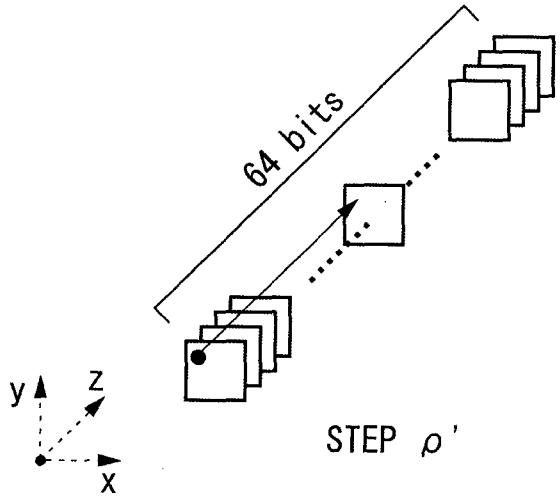
[Fig. 9B]



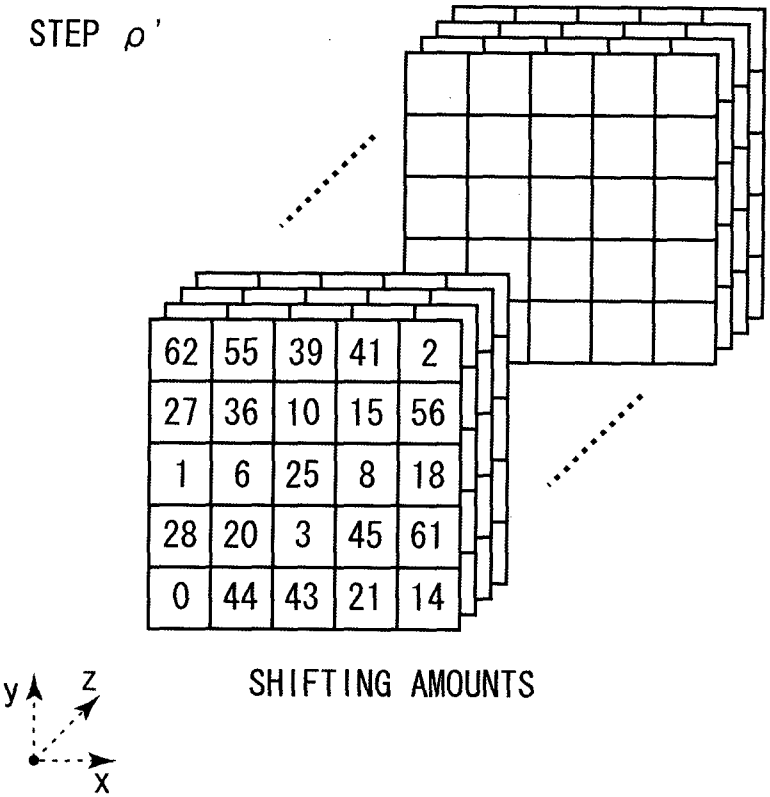
[Fig. 9C]



[Fig. 10A]



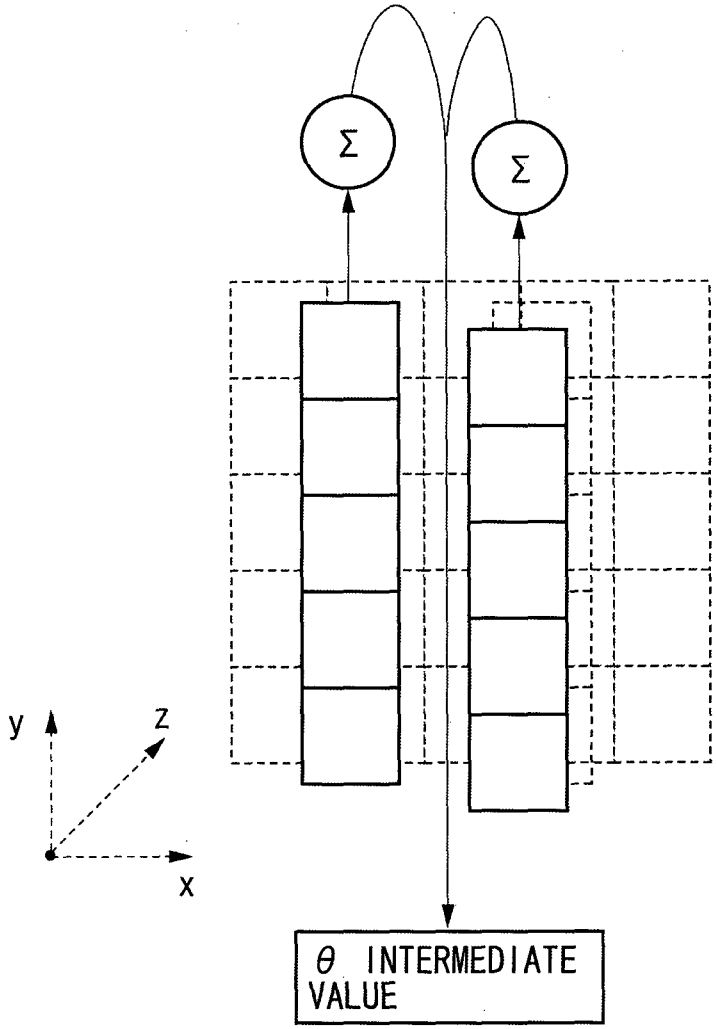
[Fig. 10B]



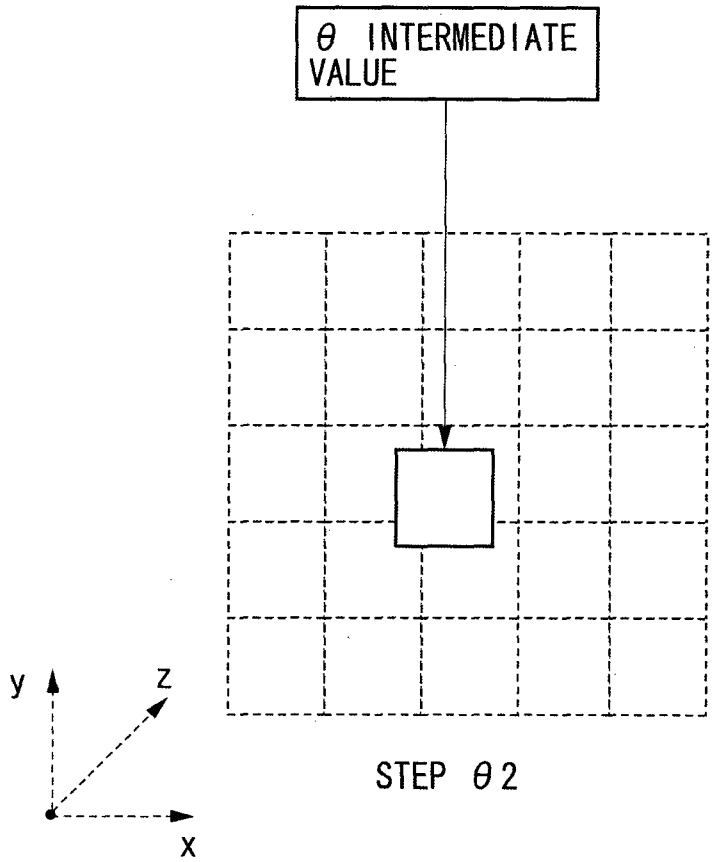
[Fig. 10C]

62	55	39	41	2
27	36	10	15	56
1	6	25	8	18
28	20	3	45	61
0	44	43	21	14

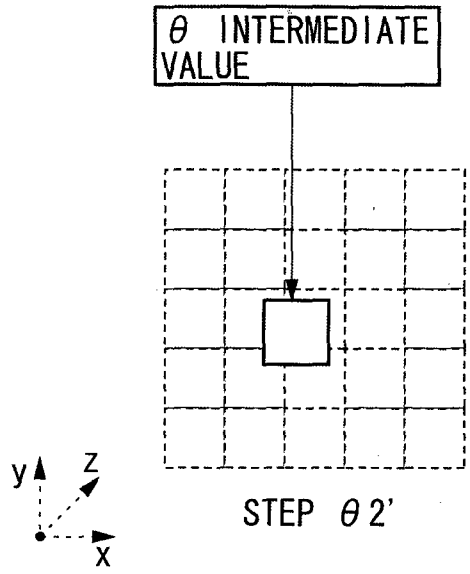
[Fig. 11]



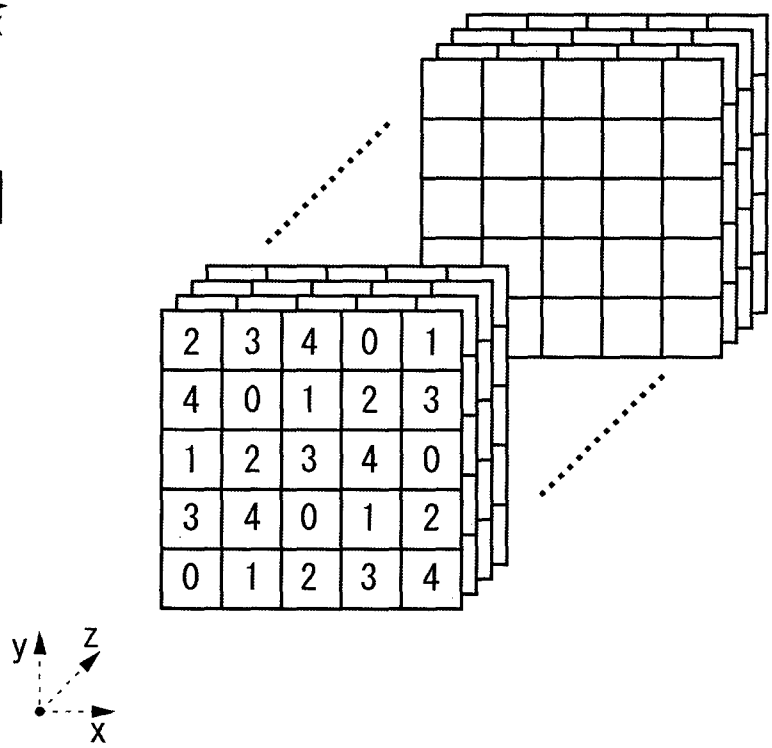
[Fig. 12]



[Fig. 13A]



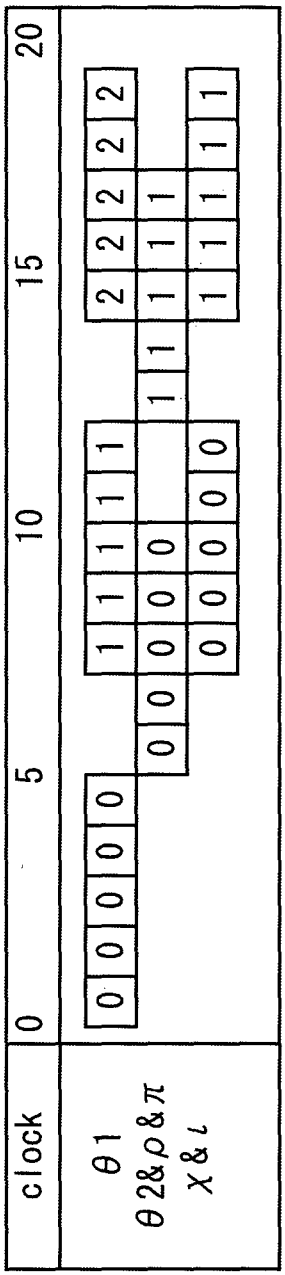
[Fig. 13B]



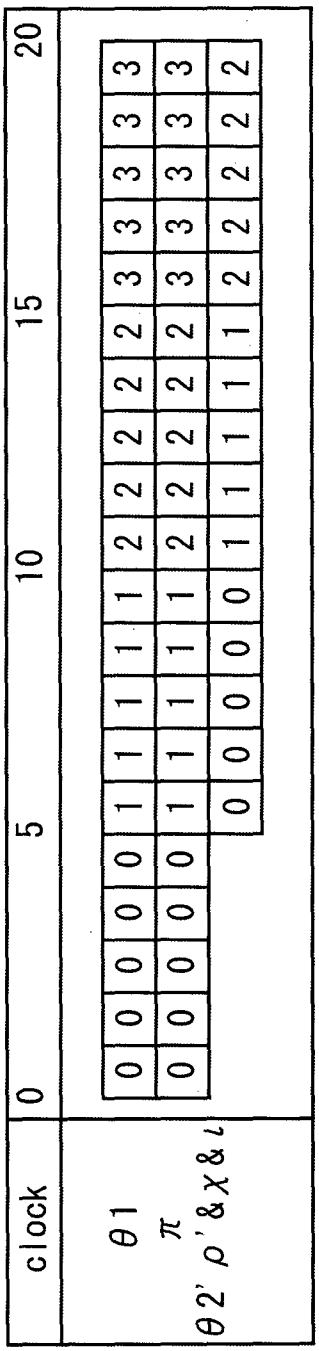
[Fig. 13C]

2	3	4	0	1
4	0	1	2	3
1	2	3	4	0
3	4	0	1	2
0	1	2	3	4

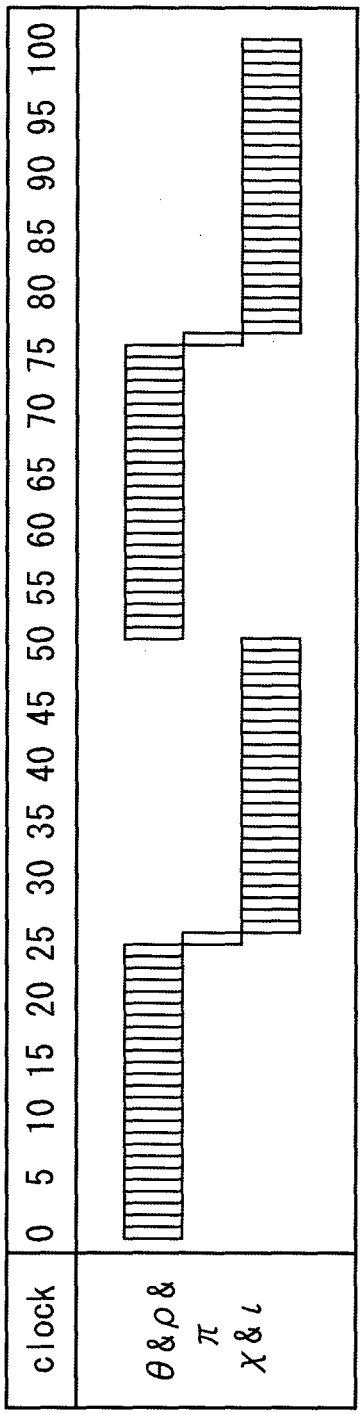
[Fig. 14A]



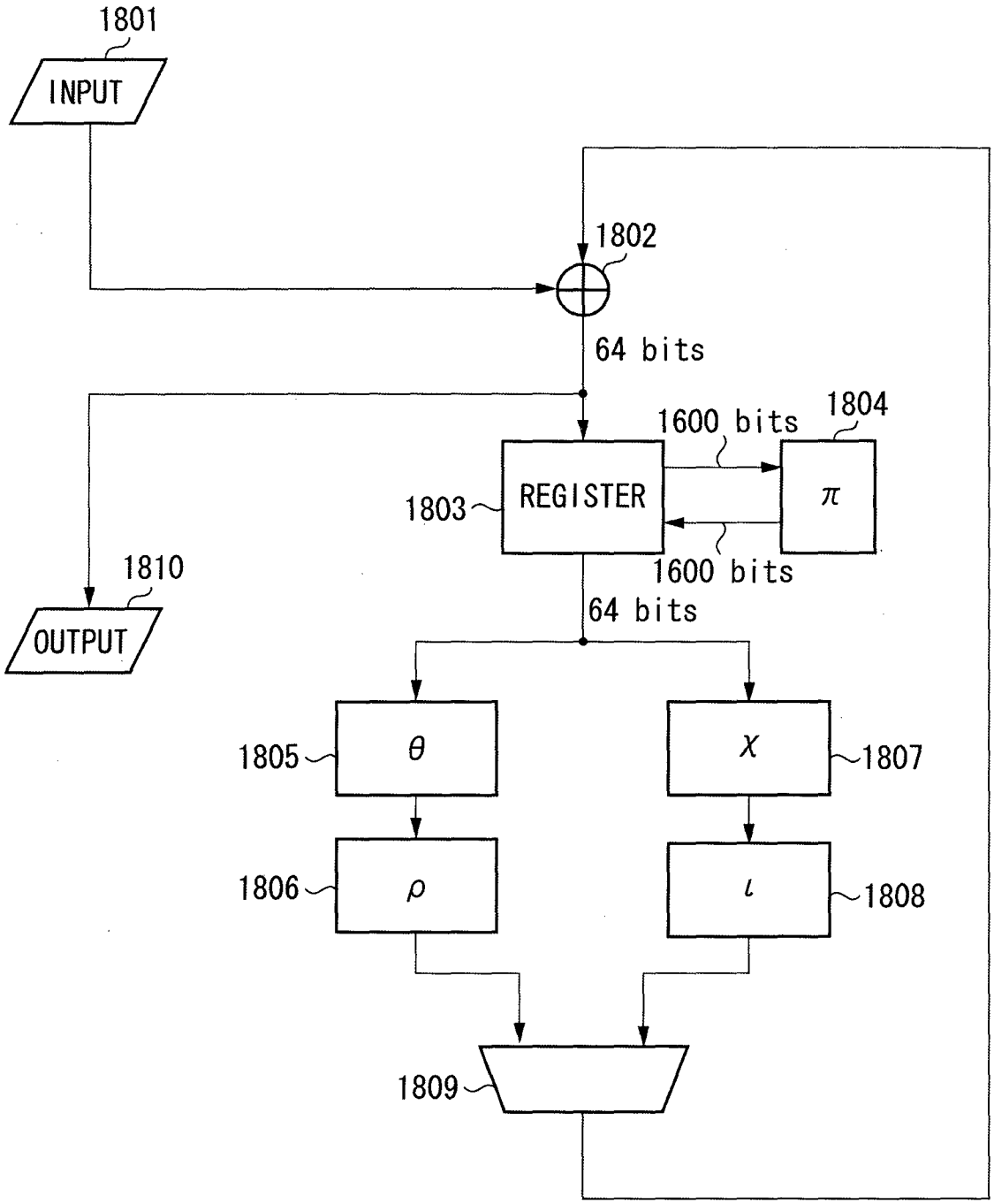
[Fig. 14B]



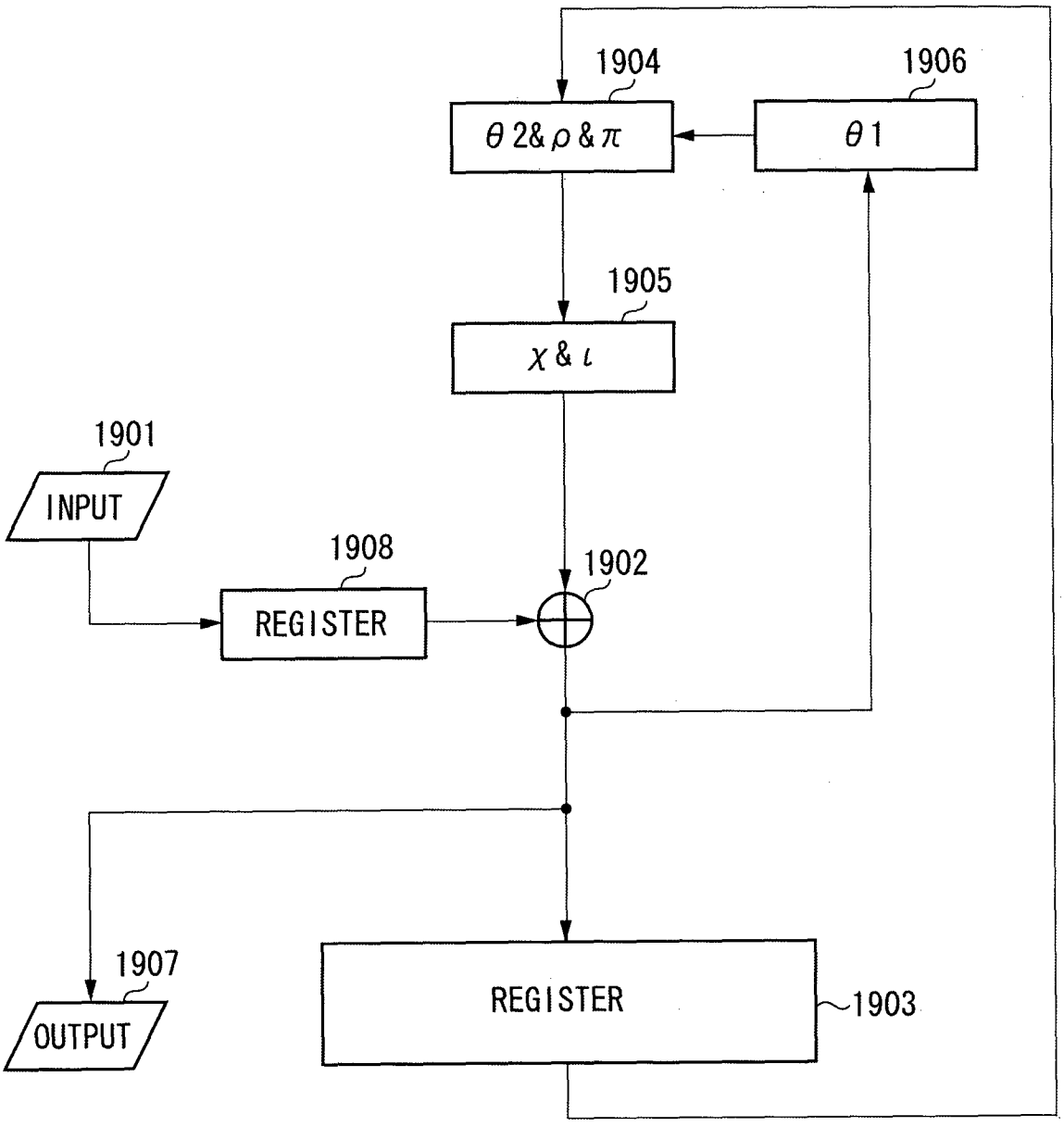
[Fig. 14C]


clock
θ&ρ&
π
χ&ι
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

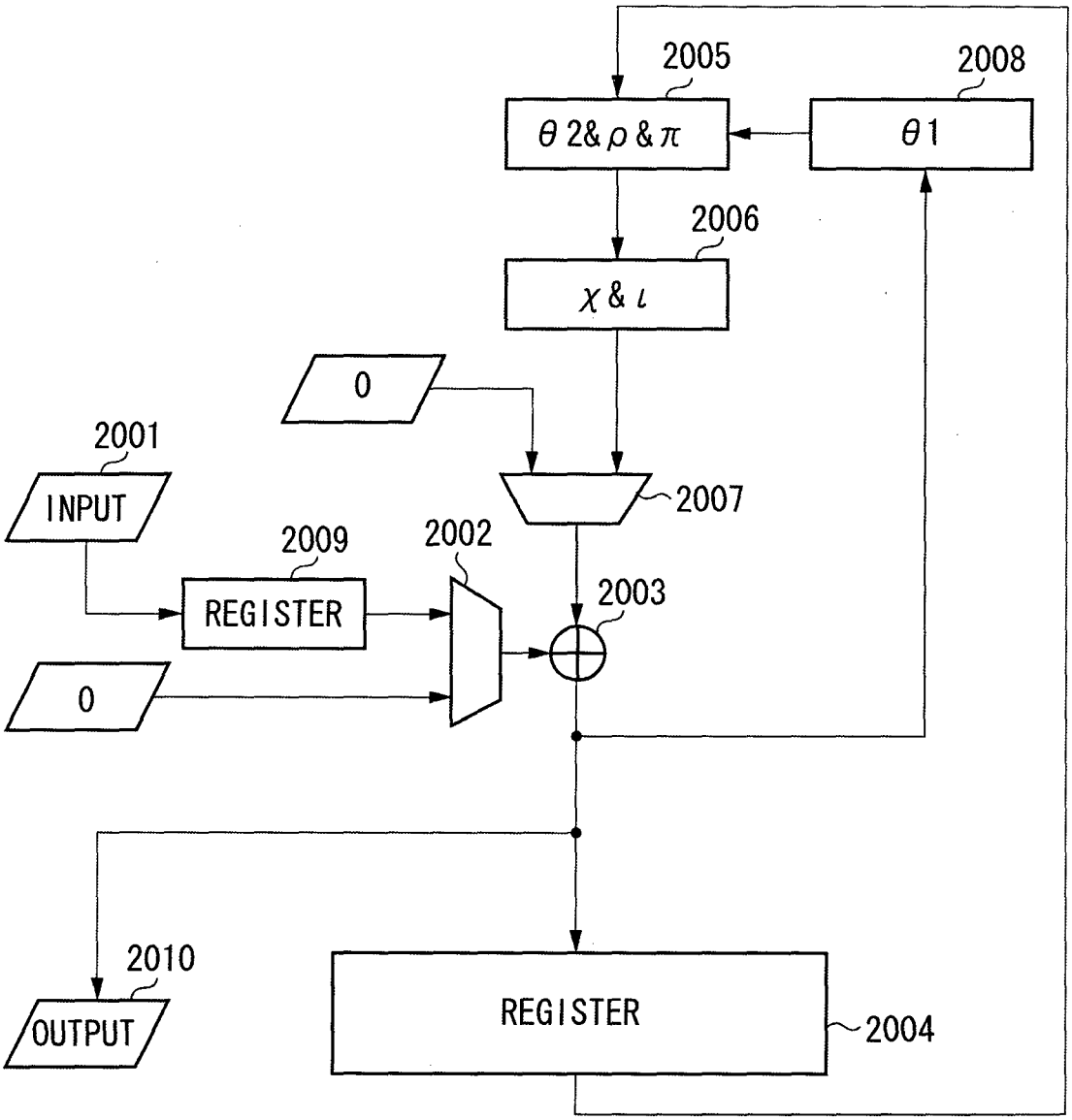
[Fig. 15]


1801
INPUT
1802
64 bits
1600 bits
1804
1803
REGISTER
π
1600 bits
1810
OUTPUT
64 bits
1805
θ
χ
1807
1806
ρ
ι
1808
1809

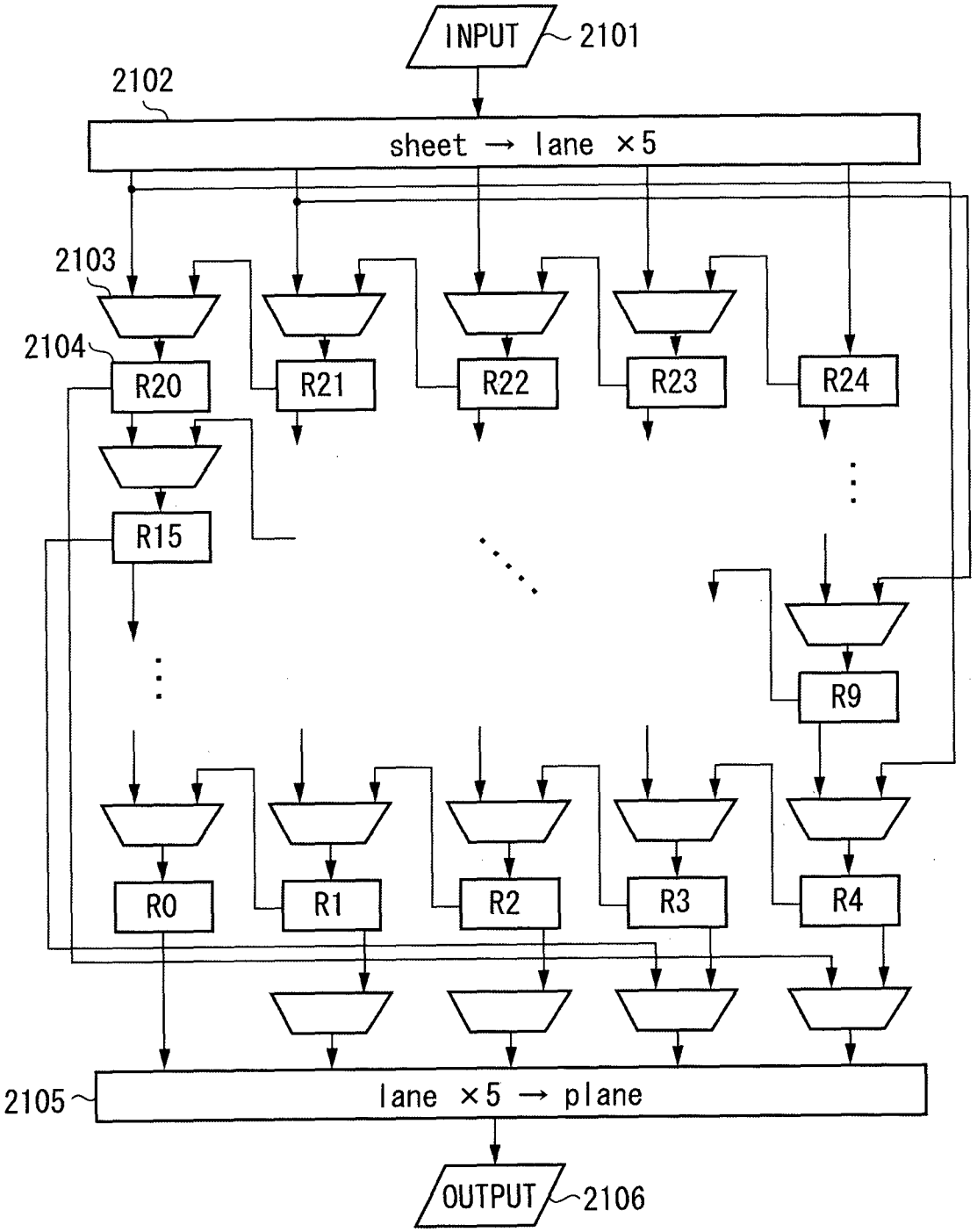
[Fig. 16]


1904
1906
θ2&ρ&π
θ1
1905
χ&ι
1901
INPUT
1908
1902
REGISTER
1907
OUTPUT
REGISTER
1903

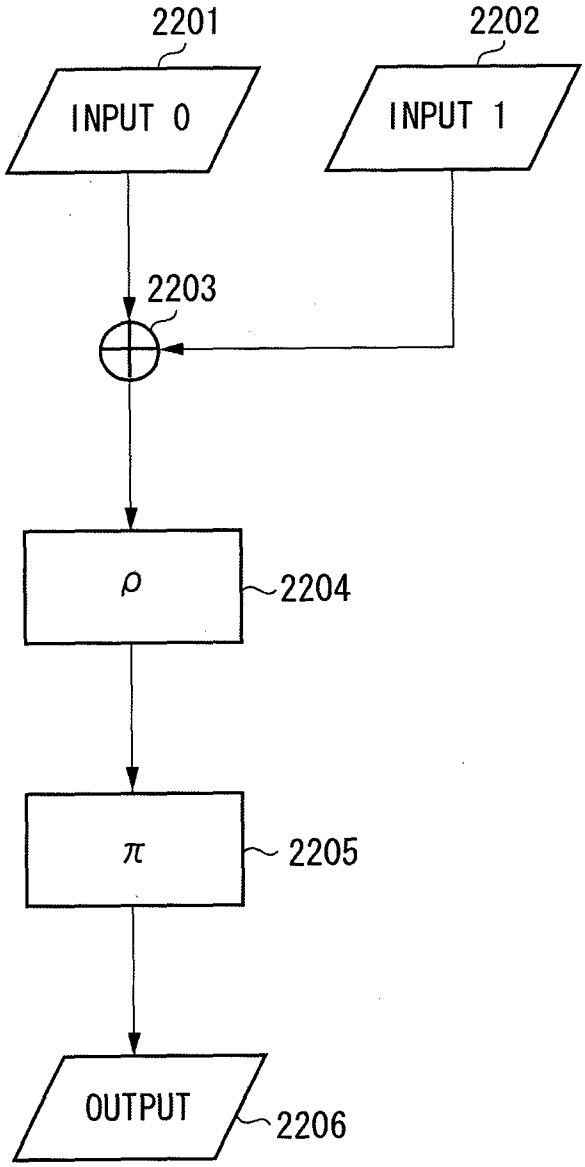
[Fig. 17]


2005
2008
θ2&ρ&π
θ1
2006
χ&ι
0
2001
INPUT
2007
2009
2002
REGISTER
2003
0
2010
OUTPUT
REGISTER
2004

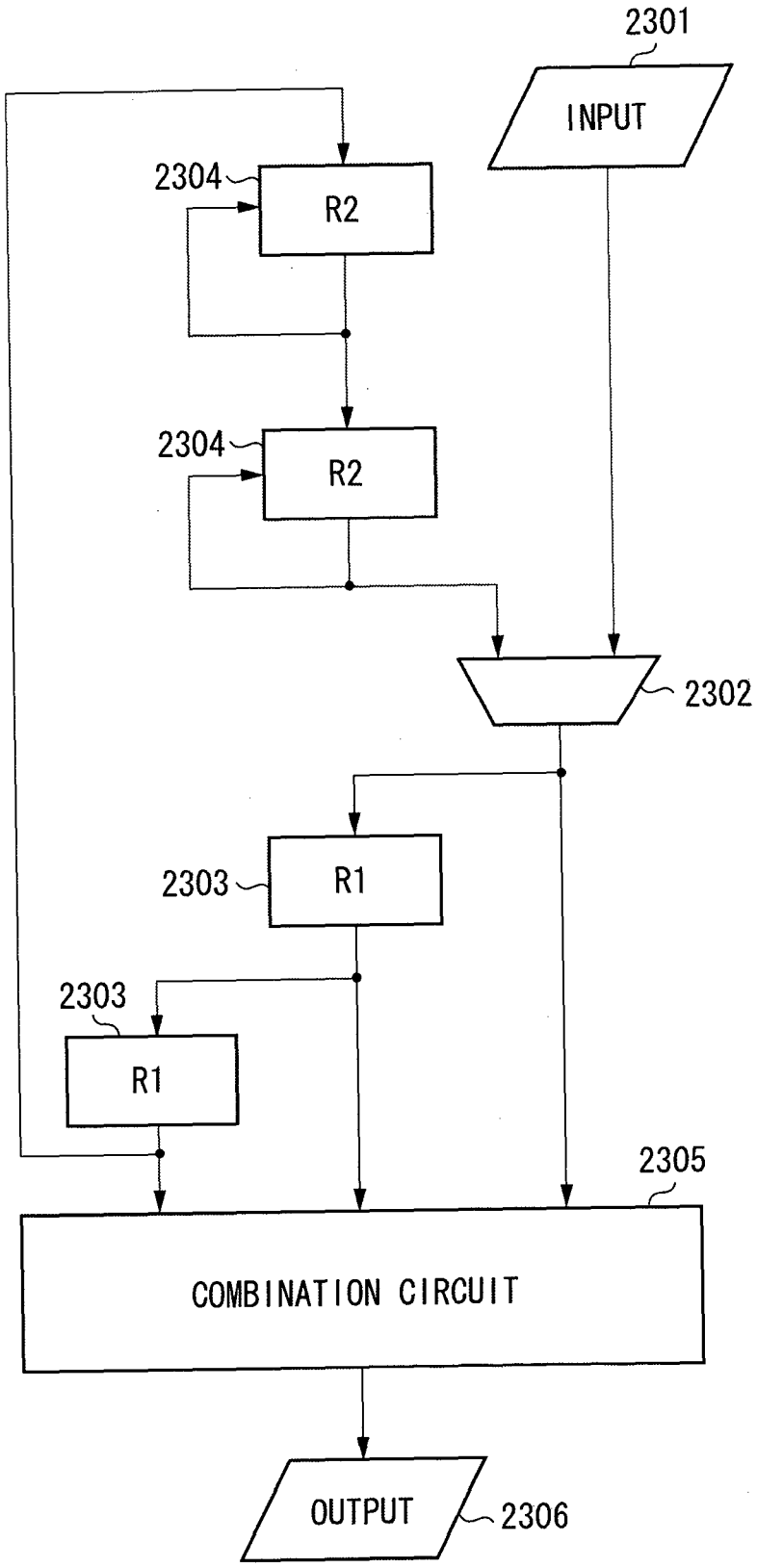
[Fig. 18]


INPUT
2101
2102
sheet → lane ×5
2103
2104
R20
R21
R22
R23
R24
R15
R9
R0
R1
R2
R3
R4
2105
lane ×5 → plane
OUTPUT
2106

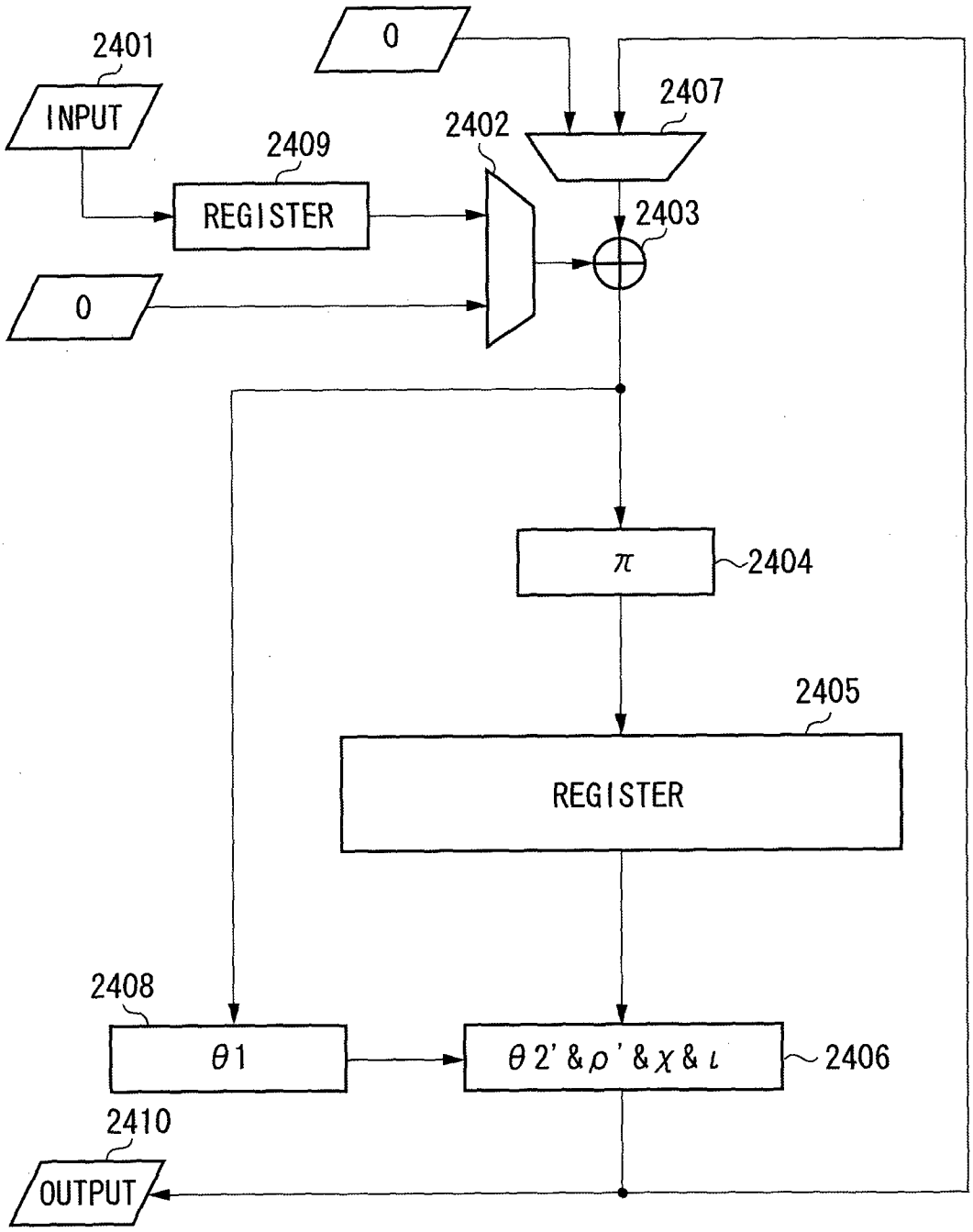
[Fig. 19]



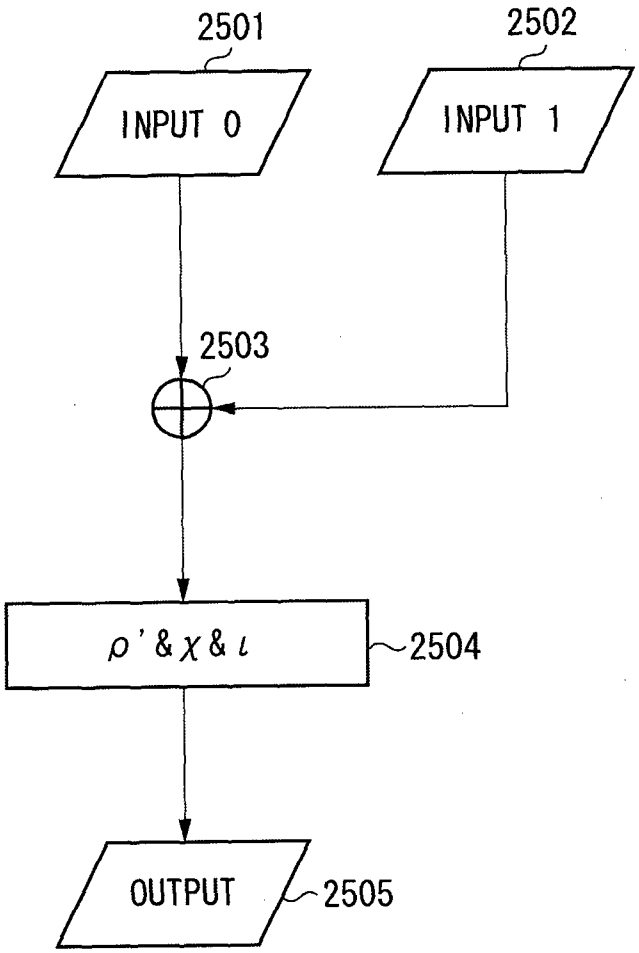
[Fig. 20]


2301
INPUT
2304
R2
2304
R2
2302
2303
R1
2303
R1
2305
COMBINATION CIRCUIT
OUTPUT
2306

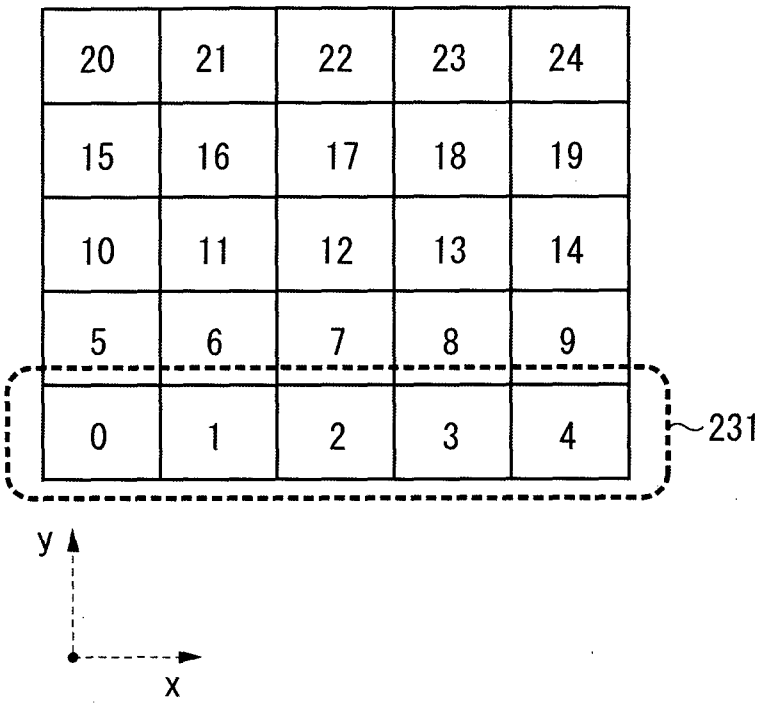
[Fig. 21]


0
2401
INPUT
2407
2402
2409
REGISTER
2403
0
π
2404
2405
REGISTER
2408
θ1
θ2'&ρ'&χ&ι
2406
2410
OUTPUT

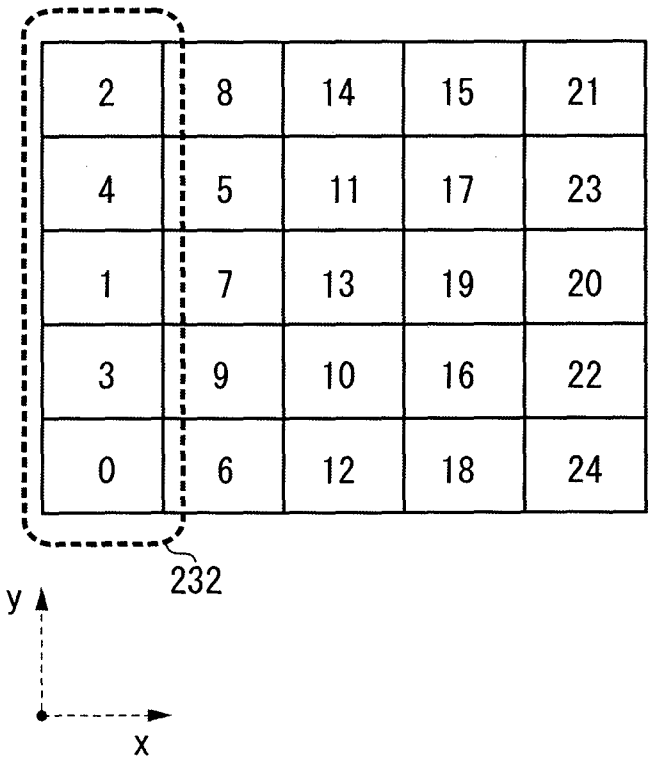
[Fig. 22]



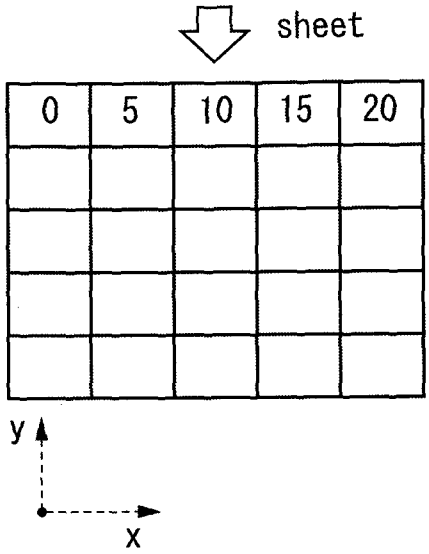
[Fig. 23A]



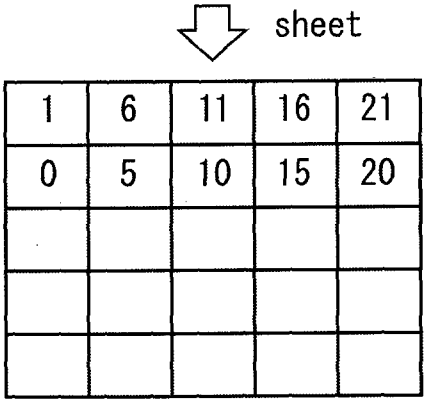
[Fig. 23B]



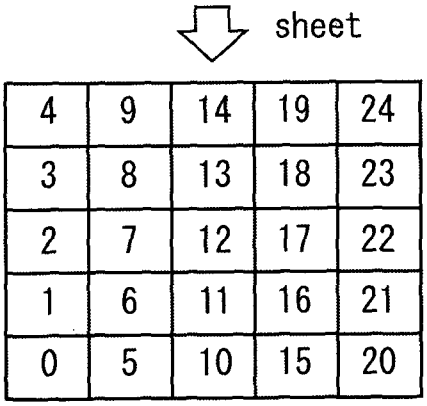
[Fig. 24A]



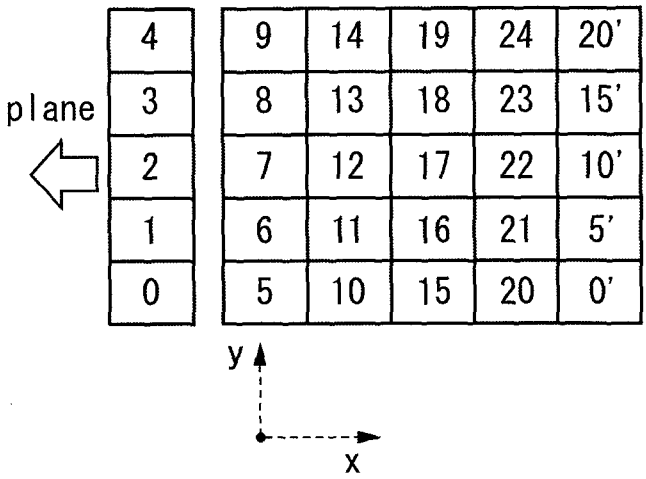
[Fig. 24B]



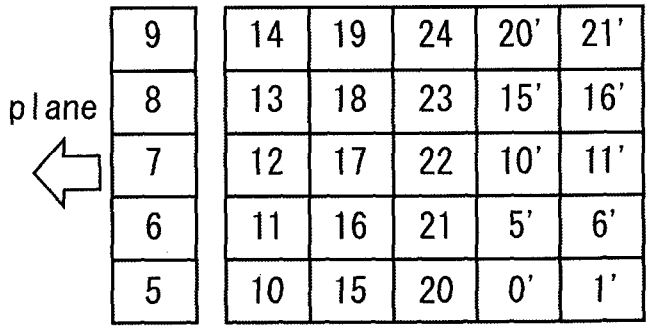
[Fig. 24C]



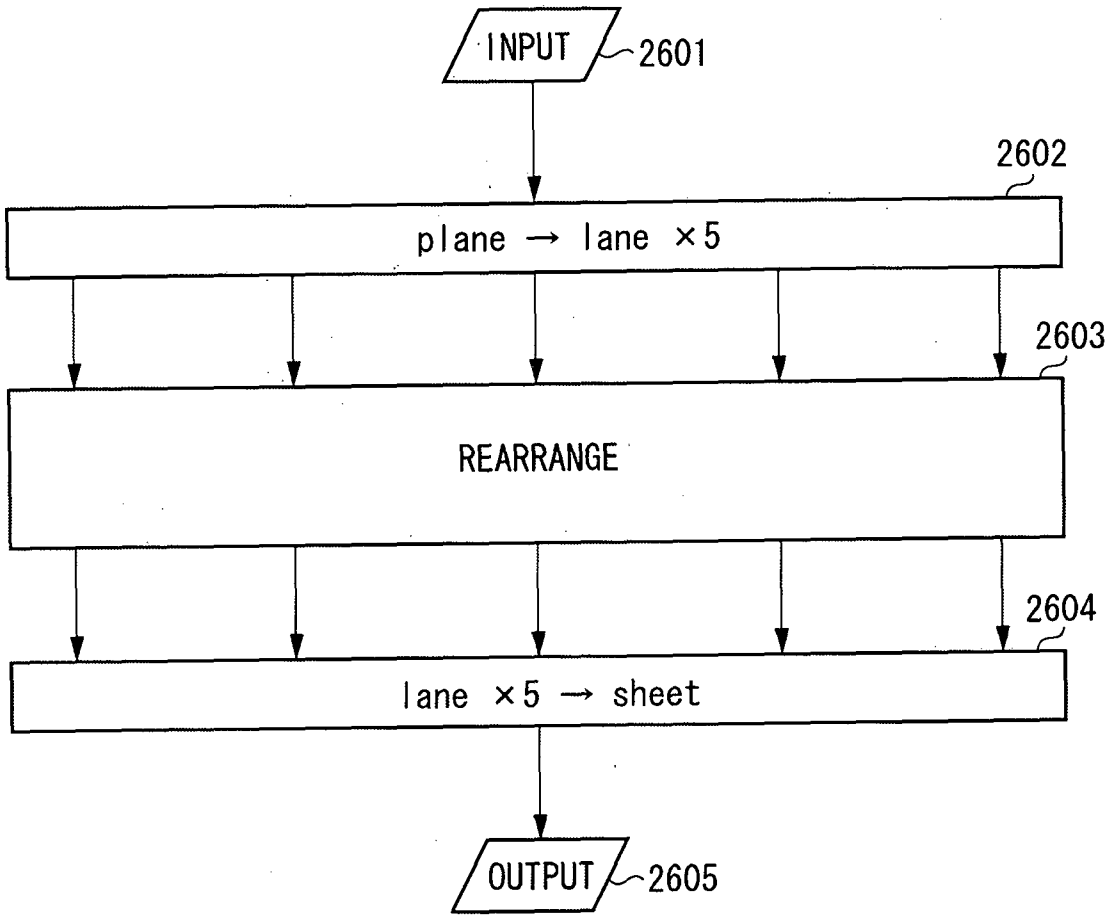
[Fig. 24D]



[Fig. 24E]



[Fig. 25]


INPUT
2601
2602
plane → lane ×5
2603
REARRANGE
2604
lane ×5 → sheet
OUTPUT
2605

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2014/054246

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G09C1/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. G09C1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2014
Registered utility model specifications of Japan 1996-2014
Published registered utility model applications of Japan 1994-2014

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

JSTPlus/JMEDPlus/JST580(JDreamIII), Science Direct, IEEE Xplore, THE ACM DIGITAL LIBRARY, Keccack, SHA-3

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Kavun, E. B. and Yalcin, T., A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications, Lecure Notes in Computer Science, Vol.6370, 2010.06, p.258-269	1-19
A	Bertoni, G. et al., KECCAK implementation overview, [online], Ver. 3.2, 2012.05.29, [retrieved on 2014-04-11], Retrieved from the Internet <URL:http://keccak.noekeon.org/files.html> especially 2.4 Plane-per-plane processing	1-19
A	Kerckhof, S. et al., Compact FPGA Implementations of the Five SHA-3 Finalists, Lecture Notes in Computer Science, Vol.7079, 2011.09, p.217-233, especially p.226-228	1-19

☑ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28.04.2014	13.05.2014

Name and mailing address of the ISA/JP	Authorized officer	
Japan Patent Office	Hiromasa Nakazato	5S 9364
3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Telephone No. +81-3-3581-1101 Ext. 3546	

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2014/054246

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Kaps, J.-P. et al., Lightweight Implementations of SHA-3 Candidates on FPGAs, Lecture Notes in Computer Science, Vol.7107, 2011.12, p.270-289, especially 3.4 Keccak	1-19
A	San, I. and At, N., Compact Keccak Hardware Architecure for Data Integrity and Authentication on FPGAs, Information Security Journal: A Global Perspective, Vol.21 No.5, 2012, p.231-242 especially 4.4 The Datapath of the Keccak Coprocessor	1-19
A	Latif, K. et al., Novel Arithmetic Architecture for High Performance Implementation of SHA-3 Finalist Keccak on FPGA Platforms, Lecture Notes in Computer Science, Vol.7199, 2012.03, p.372-378, especially 3 Implemantation	1-19
P,A	WO 2013/089682 A1 (INTEL CORPORATION) 2013.06.20, page 4, line 1 - page 8, line 6, family: none	1-19